RÉPUBLIQUE
FRANÇAISE
Liberté
Égalité
Fraternité

RECOMMANDATIONS DE SÉCURITÉ POUR UN SYSTÈME D'IA GÉNÉRATIVE

GUIDE ANSSI

ANSSI-PA-102
29/04/2024

PUBLIC VISÉ :

Développeur | Administrateur | RSSI | DSI | Utilisateur

Informations

Attention

Ce document rédigé par l'ANSSI s'intitule **« Recommandations de sécurité pour un système d'IA générative »**. Il est téléchargeable sur le site `cyber.gouv.fr`.

Sauf disposition réglementaire contraire, les recommandations n'ont pas de caractère normatif ; elles sont livrées en l'état et adaptées aux menaces au jour de leur publication. Au regard de la diversité des systèmes d'information, l'ANSSI ne peut garantir que ces informations puissent être reprises sans adaptation sur les systèmes d'information cibles. Dans tous les cas, la pertinence de l'implémentation des éléments proposés par l'ANSSI doit être soumise, au préalable, à la validation de l'administrateur du système et/ou des personnes en charge de la sécurité des systèmes d'information.

Évolutions du document :

VERSION	DATE	NATURE DES MODIFICATIONS
1.0	29/04/2024	Version initiale

Table des matières

1
Contexte

1.1 Introduction

Si le thème de l'intelligence artificielle (IA) existe depuis longtemps dans le domaine de la recherche, les possibilités offertes par les puissances de calcul et le traitement de données massives ont permis d'ouvrir de nouvelles opportunités. Parmi celles-ci, on note un essor important de produits permettant de générer une réponse à une question formulée en langage naturel à partir d'un modèle entraîné sur des volumes de données très importants. Ces modèles d'IA sont généralement appelés *Large Language Model* (LLM) et entrent dans la catégorie de l'*IA générative* (voir les définitions en section 1.2).

Le récent engouement concernant ces produits et services, dont certains sont rendus facilement accessibles au grand public, a entraîné des réflexions au sein des organisations (entreprises, administrations), afin d'étudier les éventuels gains de productivité qui pourraient en découler.

Si cette technologie offre de nouvelles perspectives dans l'organisation du travail, il convient d'adopter une posture de vigilance et de prudence lors de son déploiement et de son intégration dans un système d'information existant. En effet, le déploiement d'outils d'IA générative engendre de nouvelles menaces pouvant avoir un impact conséquent, par exemple sur la confidentialité des données qu'ils traitent, mais également sur l'intégrité des systèmes d'information avec lesquels ils sont connectés.

Ce document a ainsi pour objet de donner des recommandations de sécurité sur la mise en œuvre de solutions d'IA générative reposant sur des LLM au sein d'entités publiques et privées.

1.2 Définitions

IA générative

L'IA générative est un sous-ensemble de l'intelligence artificielle, axé sur la création de modèles qui sont entraînés à générer du contenu (texte, images, vidéos, etc.) à partir d'un corpus spécifique de données d'entraînement.

Large Language Model

Catégorie de modèles d'IA générative qui peuvent générer du texte proche du langage naturel d'un être humain, et qui sont généralement entraînés sur un large ensemble de données.

Modèle d'IA

Un modèle d'IA désigne, dans le contexte de ce guide, un réseau de neurones et ses paramètres (poids, biais [1]).

Système d'IA

Un système d'IA englobe l'ensemble des composants techniques d'une application reposant sur un modèle d'IA : l'implémentation de ce modèle d'IA, les services frontaux pour les utilisateurs, les bases de données, la journalisation, etc.

Requête

Une requête (ou *prompt*) désigne l'instruction sous forme de texte envoyée par l'utilisateur au système d'IA.

Attaque adverse

Une attaque adverse (*adversarial attack*), parfois aussi appelée « attaque antagoniste » ou « attaque par exemples contradictoires » vise à envoyer à un système d'IA une ou plusieurs requêtes malveillantes dans le but de tromper ou d'altérer son bon fonctionnement.

1. Dans un réseau de neurones, un poids est un coefficient de puissance de la connexion entre 2 neurones, qui s'ajuste pendant toute la phase d'entraînement. Un biais est une constante liée à un neurone permettant une « compensation » dans le calcul du résultat.

1.3 Périmètre

Ce document traite principalement des cas d'usages suivants :

- la synthèse ou le résumé d'un corpus documentaire ;
- l'extraction d'informations ou la génération de texte à partir d'un corpus documentaire ;
- les agents conversationnels [2] (appelés aussi *Chatbot*) ;
- la génération de code source pour les développeurs d'applications.

Le corpus documentaire identifié peut être « multimodal », c'est-à-dire qu'il peut impliquer diverses catégories de données en entrée : textes, images, sons, vidéos, etc. En revanche, le guide se focalise principalement sur la génération textuelle en sortie et ne traite pas particulièrement de la génération d'image ou de vidéo (même si la majorité des recommandations sont applicables à ces cas d'usages).

Ce corpus documentaire intègre les données d'entraînement du modèle, mais peut aussi s'appuyer sur des données additionnelles ou des documents fournis directement en entrée par l'utilisateur.

Ce document ne traite que de la sécurisation d'une architecture de système d'IA générative reposant sur un LLM.

Les problématiques de sécurité liées à la *qualité* [3] des données et à la *performance* [4] d'un modèle d'IA ne sont pas traitées dans ce document.

De même, si d'autres enjeux comme l'éthique, la vie privée, la propriété intellectuelle, la protection du secret des affaires ou encore la protection des données personnelles sont également des enjeux à prendre en compte dans la conception d'un modèle d'IA, ces derniers ne sont pas du domaine d'expertise de l'ANSSI et ne sont donc pas abordés dans ce guide.

Pour l'ensemble de ces sujets, il est possible de prendre connaissance des travaux de l'ENISA [6, 7], du BSI [1], du NIST [15, 16] ou encore de la CNIL [2].

L'ANSSI a également cosigné un document du NCSC-UK [14] sur la sécurisation de l'IA en novembre 2023.

2. Un agent conversationnel est défini ici comme une application permettant un échange écrit entre l'utilisateur et le système d'IA et non un échange oral.

3. La qualité des données désigne généralement un critère plutôt métier. Des critères de qualité des données d'un point de vue métier peuvent être par exemple l'origine, la quantité, l'exhaustivité, la pertinence, l'exactitude, la représentativité (au sens statistique), ou encore le respect d'une structure donnée.

4. La performance d'un modèle d'IA est également un concept métier très dépendant des objectifs fixés lors de la conception du modèle. Elle peut inclure plusieurs facteurs comme la précision, la pertinence ou encore la rapidité des réponses générés pour les utilisateurs par exemple.

2
Synthèse

La mise en œuvre d'un système d'IA générative peut se décomposer en 3 phases cycliques : une première phase d'entraînement du modèle d'IA à partir de données spécifiquement choisies, puis une phase d'intégration et de déploiement, et enfin une phase de production opérationnelle dans laquelle les utilisateurs peuvent accéder au modèle d'IA entraîné, par l'intermédiaire du système d'IA.

Ces 3 phases doivent chacune faire l'objet de mesures de sécurisation spécifiques, qui dépendent en partie du choix de sous-traitance retenu pour chaque composante (hébergement, entraînement du modèle, tests de performance, etc.) ainsi que de la sensibilité des données utilisées à chaque étape et de la criticité du système d'IA dans sa finalité.

En complément des menaces classiques inhérentes à tout système d'information, un système d'IA générative peut-être soumis à des attaques spécifiques visant par exemple à perturber le bon fonctionnement de celui-ci (attaques adverses) ou bien à exfiltrer des données traitées par celui-ci.

La question de la protection des données, notamment des données d'entraînement, est donc un enjeu essentiel d'un système d'IA générative, avec comme corollaire la problématique du besoin d'en connaître des utilisateurs lorsqu'ils interrogent le modèle. En effet, ce dernier est conçu pour générer une réponse à partir de l'ensemble des données auxquelles il a eu accès lors de l'entraînement, ainsi que des données additionnelles qui peuvent être issues de sources internes sensibles.

L'utilisation d'un système d'IA générative doit donc répondre à des besoins de confidentialité (*il faut à ce titre rappeler que l'envoi de données sensibles à des outils grand public sur Internet* [5] *est à proscrire*) mais également des besoins en intégrité et en disponibilité. Les interactions du système d'IA avec d'autres applications ou composants du SI doivent ainsi être sécurisées, limitées au strict besoin opérationnel, et doivent pouvoir être contrôlées par un humain lorsque celles-ci sont critiques pour l'organisation.

Certains usages spécifiques, comme l'assistance par IA pour le développement d'applications, soulèvent des problématiques importantes et doivent donc être cadrés (en ayant une grande vigilance sur des modules ou des applications sensibles), contrôlés par des humains et testés régulièrement (avec des outils automatiques d'analyse de code source).

Enfin, la protection des modèles d'IA peut constituer un enjeu au même titre que la protection des données, non seulement pour des raisons de protection du potentiel scientifique et technique de la nation (recherche académique, modèles utilisés pour la sécurité nationale, etc.), mais aussi parce qu'un attaquant ayant connaissance de l'architecture et des paramètres d'un modèle peut potentiellement améliorer ses capacités d'attaques à d'autres fins (exfiltration de données, etc.).

5. Comme par exemple *ChatGPT*, *Gemini* ou encore *DeepL* pour la traduction.

3
Description d'un système d'IA générative

Attention

Le cycle de vie et l'architecture présentés dans ce chapitre sont donnés à titre d'exemples pour faciliter la compréhension des recommandations. Ils n'ont donc pas de vocation normative. En particulier, le séquencement des fonctions présentées peut varier et, selon les cas d'usages, ces fonctions ne sont pas toujours implémentées dans un système d'IA.

3.1 Cycle de vie d'un système d'IA générative

La figure 1 décrit un exemple de cycle de vie d'un système d'IA générative.

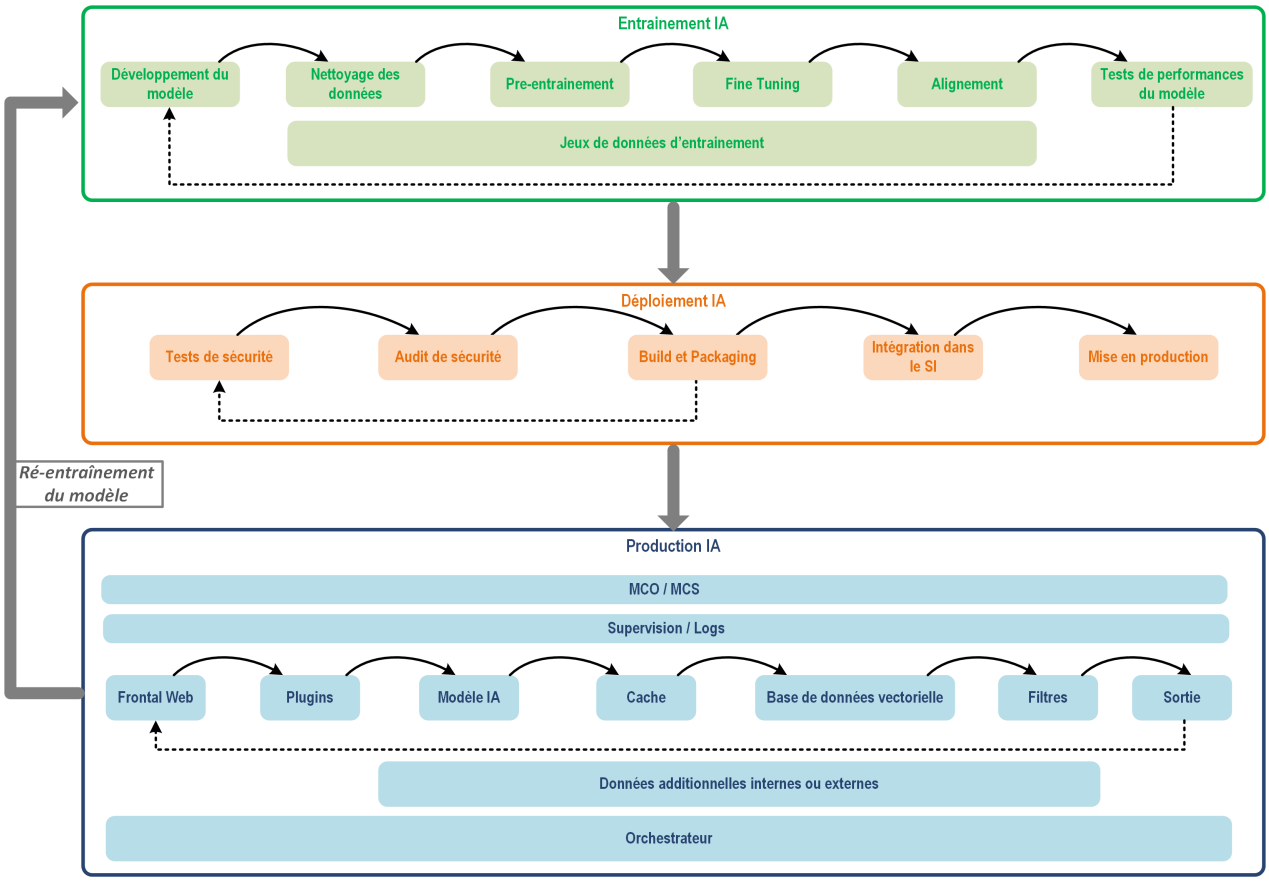

FIGURE 1 – Exemple de cycle de vie d'un système d'IA générative

Les 3 phases d'entraînement, de déploiement et de production [6] impliquent potentiellement des environnements différents et des utilisateurs différents. Il est important que ces 3 phases du cycle de vie d'un système d'IA générative fassent chacune l'objet d'une attention particulière du point de vue de la sécurité.

Ces 3 phases peuvent être réalisées dans des environnements distincts, par exemple la phase d'entraînement dans un *Cloud* public et la phase de déploiement et de production en interne de l'entité. Néanmoins, des mesures de sécurisation adaptées doivent s'appliquer quel que soit l'environnement choisi.

Le ré-entrainement d'un modèle d'IA présenté dans ce schéma n'implique généralement pas de refaire toutes les étapes présentées dans la phase d'entraînement (très souvent, seules les étapes de *fine-tuning* ou d'alignement sont réalisées).

La figure 2 présente des exemples de partage de responsabilités dans l'ensemble des phases de conception d'un système d'IA générative.

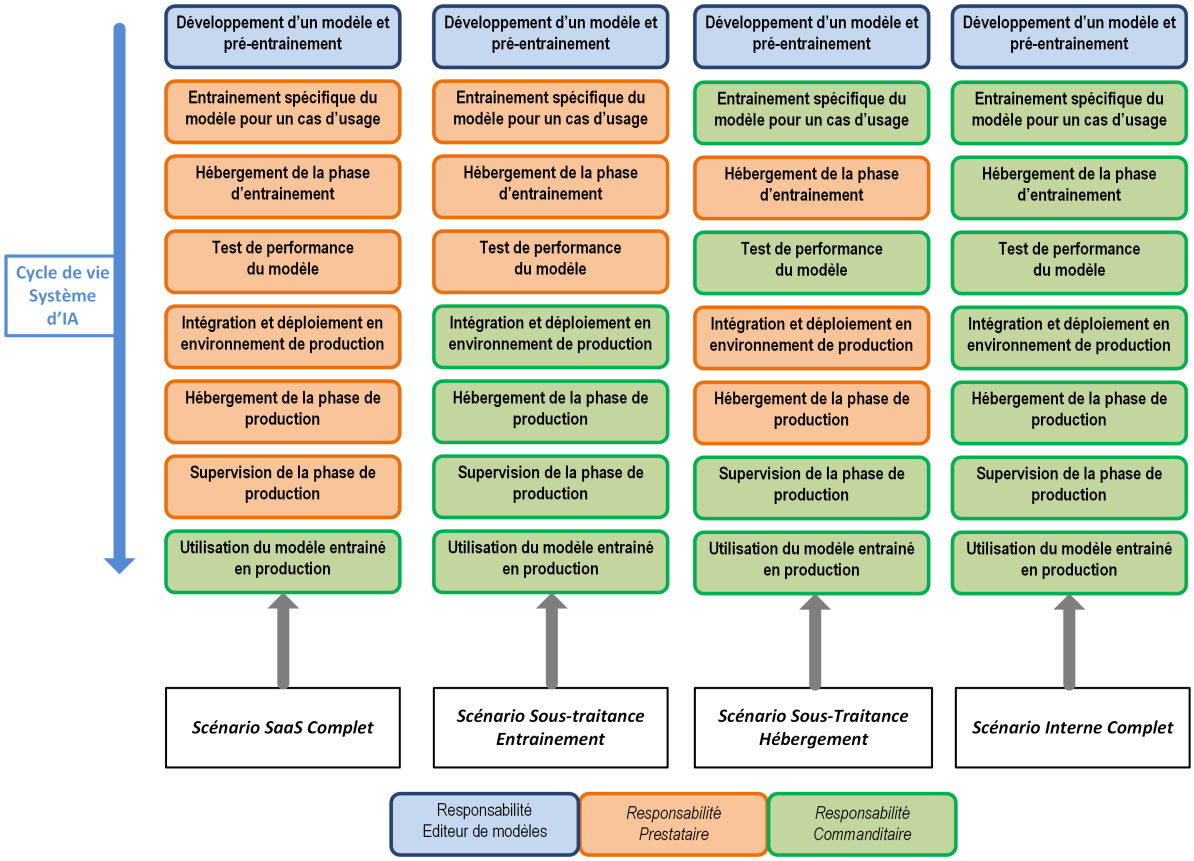

FIGURE 2 – Scénarios de partage de responsabilités d'un système d'IA générative

Les risques et les impacts du point de vue de la sécurité seront évaluées en fonction du scénario retenu par l'organisation.

6. Cette phase de production peut parfois être appelée phase d'inférence du modèle d'IA, c'est-à-dire que le modèle réalise des prédictions pour des utilisateurs donnés.

La figure 3 décrit l'intégration d'un système d'IA générative dans un SI et les points d'attention à prendre en compte en ce qui concerne les interactions internes et externes.

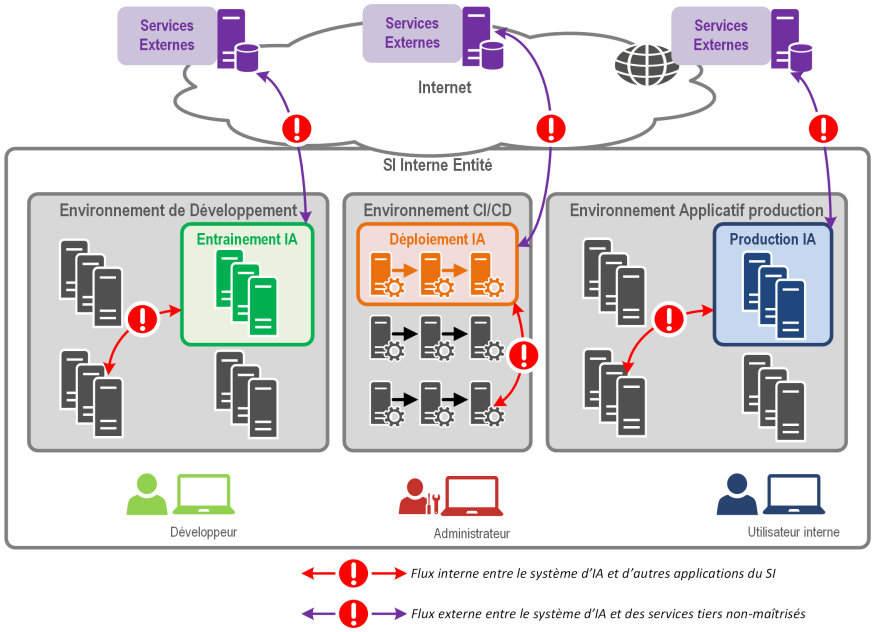

FIGURE 3 – Intégration d'un système d'IA générative dans un SI existant

Ces interactions doivent faire l'objet d'une attention particulière et doivent être intégrées dans le périmètre de l'analyse à toutes les phases du projet.

R1

Intégrer la sécurité dans toutes les phases du cycle de vie d'un système d'IA

Des mesures de sécurité doivent être identifiées et appliquées dans chacune des 3 phases du cycle de vie d'un système d'IA : entraînement, déploiement et production. Ces mesures dépendent fortement du scénario de partage de responsabilités retenu et de la sous-traitance associée. Elles doivent également tenir compte des interactions avec d'autres applications ou composants internes ou externes au SI.
Il est possible de se référer au guide d'hygiène de l'ANSSI [17] pour disposer d'un socle de base de sécurité à appliquer.

3.2 Architecture d'un système d'IA générative

La figure 4 décrit un exemple d'architecture générique d'un système d'IA générative.

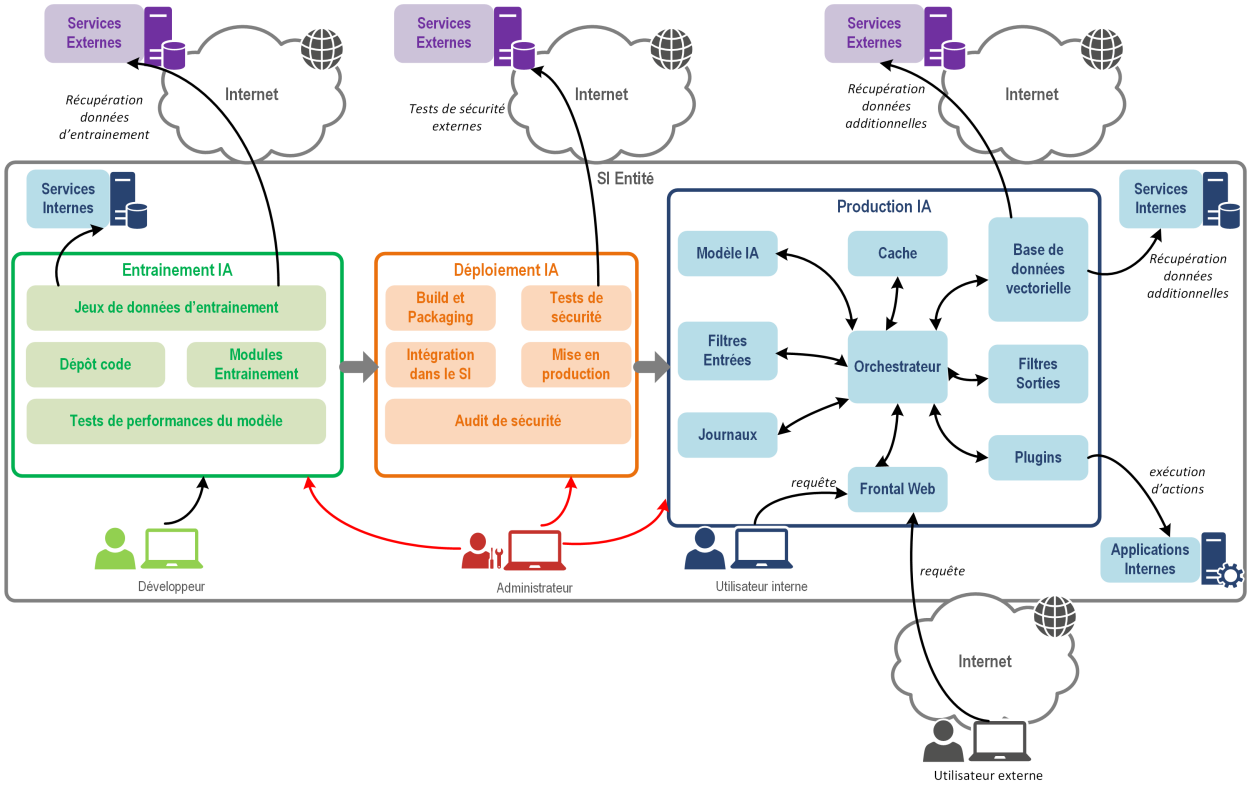

FIGURE 4 – Exemple d'architecture générique d'un système d'IA générative

Cette architecture ne reprend pas de manière exhaustive l'ensemble des composants d'un système d'IA générative mais a pour but d'identifier les chemins d'attaques potentiels ciblant une entité.

Plusieurs éléments sont importants à relever dans ce schéma :

- les différentes populations qui peuvent accéder à un système d'IA selon les phases : utilisateurs, développeurs, administrateurs, auditeurs, etc. ;
- la base de données vectorielle, qui est généralement utilisée pour stocker sous forme de vecteurs des données d'indexation de données additionnelles, dans le but d'enrichir [7] les requêtes des utilisateurs avant l'envoi au modèle (concept appelé RAG - *Retrieval Augmentation Generation*). Cette base de données peut être construite à partir de sources de données internes à l'organisation ou externes depuis des sources partenaires ;
- les filtres en entrée et sortie du modèle d'IA, permettant d'assurer une défense en profondeur contre des requêtes malveillantes ou les comportements non-souhaités du système d'IA ;
- les *plugins* ou composants additionnels, qui peuvent être utilisés pour connecter le système d'IA à d'autres ressources métier ou techniques de l'entité ou extérieures à l'entité.

7. Une base de données vectorielle est particulièrement utilisée dans le contexte des LLM car elle peut permettre d'établir des comparaisons, d'identifier des relations entre objets, et ainsi de comprendre le contexte.

4
Scénarios d'attaques sur l'IA générative

Un système d'IA générative est de prime abord une application métier standard, qui doit disposer du même socle de sécurité que toute autre application métier de l'entité. Toutefois, en complément de ce socle de sécurité, l'entité doit prendre en compte des menaces spécifiques à un système d'IA générative.

Ces menaces peuvent être déclinées en 3 grandes catégories d'attaques [8] :

- **Attaques par manipulation** : ces attaques consistent à détourner le comportement du système d'IA en production au moyen de requêtes malveillantes. Elles peuvent provoquer des réponses inattendues, des actions dangereuses ou un déni de service ;
- **Attaques par infection** : ces attaques consistent à contaminer un système d'IA lors de sa phase d'entraînement, en altérant les données d'entraînement ou en insérant une porte dérobée ;
- **Attaques par exfiltration** : ces attaques consistent à dérober des informations sur le système d'IA en production, comme les données ayant servi à entraîner le modèle, les données des utilisateurs ou bien des données internes du modèle (paramètres).

Dans le contexte de l'IA générative, ces attaques peuvent porter atteinte aux besoins de sécurité suivants :

- **Confidentialité** : l'objectif est de protéger un système d'IA contre la fuite d'informations considérées comme sensibles : jeux de données d'entraînement, requêtes des utilisateurs, paramètres des modèles, données additionnelles internes, etc. ;
- **Intégrité** : l'objectif est de protéger un système d'IA contre une modification non prévue de son comportement. L'intégrité peut concerner directement le modèle (paramètres) ou bien viser les jeux de données d'entraînement (empoisonnement) ou encore les composants techniques permettant le bon fonctionnement du système d'IA : scripts [9], bibliothèques externes (*supply-chain attack*), configurations des services, etc. ;
- **Disponibilité** : l'objectif est de protéger un système d'IA contre des dénis de service ou des actions visant à dégrader ses performances (requêtes malveillantes) ;
- **Traçabilité** : l'objectif est de garantir d'une part l'explicabilité [10] et l'imputabilité des actions réalisées sur un système d'IA. Ces éléments peuvent faciliter le travail d'investigation et de remédiation après un incident de sécurité.

8. Ces catégories sont reprises de la taxonomie de la CNIL à ce sujet : https://linc.cnil.fr/petite-taxonomie-des-attaques-des-systemes-dia.

9. Ces scripts peuvent être par exemple des scripts de *fine-tuning* du modèle d'IA ou bien des scripts de déploiement ou de maintenance informatique du système d'IA.

10. Comme défini par la CNIL, l'explicabilité est la capacité de mettre en relation et de rendre compréhensible les éléments pris en compte par le système d'IA pour la production d'un résultat.

La figure 5 décrit quelques exemples d'attaques sur un système d'IA générative dans un SI.

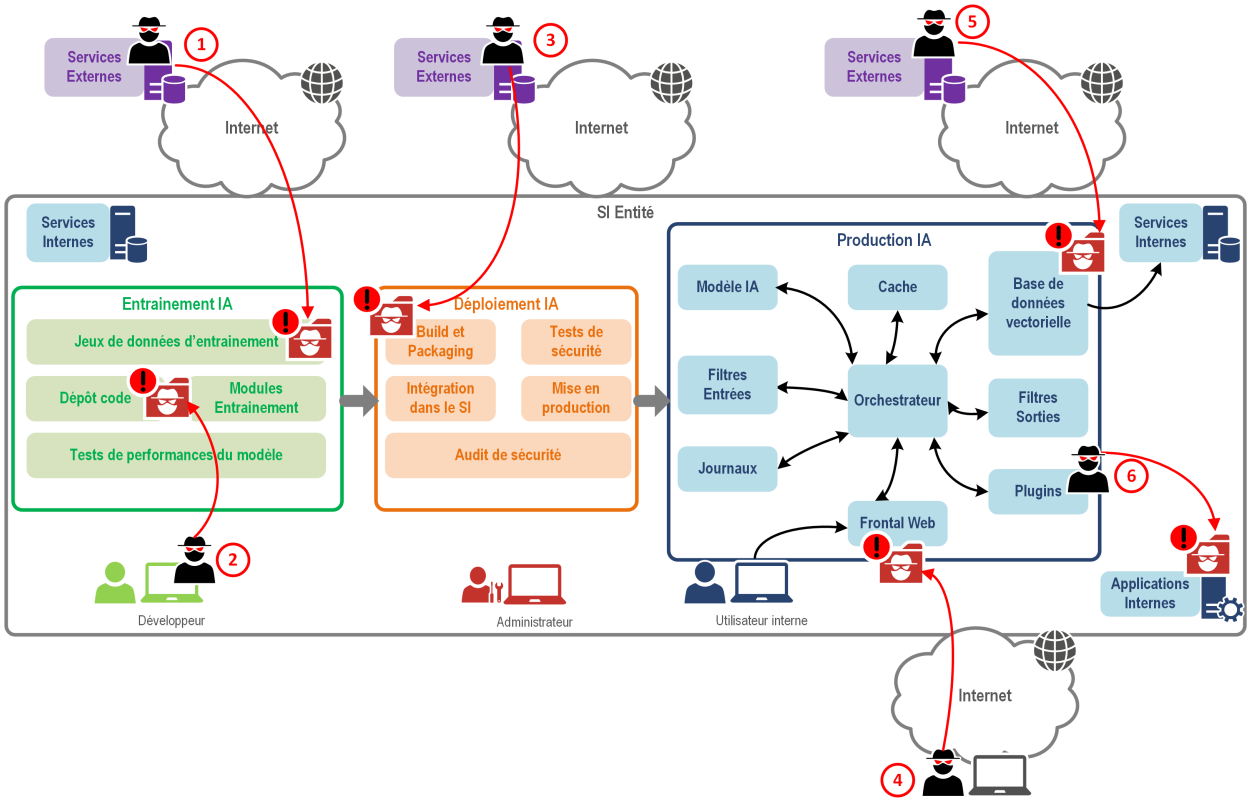

FIGURE 5 – Scénarios d'attaques sur un système d'IA générative dans un SI

1. l'attaquant dispose d'un accès à une source de données et réalise au préalable un empoisonnement des données utilisées pour entraîner le modèle d'IA, ce qui permet de détourner son usage une fois le système d'IA en production (ex. : déclencher une action malveillante de manière cachée à partir d'une requête spécifique) ;
2. l'attaquant dispose d'un accès à l'environnement de développement et insère une porte dérobée dans le code du système d'IA (ex. : altérer directement les paramètres du modèle ou bien la configuration d'un composant technique du système d'IA) ;
3. l'attaquant dispose d'un accès à un service externe de test avant déploiement et détourne le processus d'intégration (ex. : envoyer un résultat trompeur ou malveillant à la chaîne d'intégration) ;
4. l'attaquant utilise une technique d'attaque adverse pour exfiltrer des données sensibles traitées par le modèle d'IA (ex. : récupérer des données d'entraînement ou des requêtes d'autres utilisateurs du service) ou bien il effectue des requêtes malveillantes pour provoquer un déni de service ;
5. l'attaquant dispose d'un accès à une ressource externe accédée par le système d'IA et envoie une réponse malveillante qui est intégrée par le modèle (ex. : il envoie une URL qui pointe sur un site Web malveillant qu'il contrôle) ;
6. l'attaquant dispose d'un accès à un *plugin* utilisé par le système d'IA et injecte des commandes malveillantes lors d'une action vers une application métier (ex. : insérer du code malveillant dans le corps d'un mail généré par le système d'IA).

Dans le contexte de ce guide (IA générative LLM), les impacts suivants peuvent être identifiés :

- ternissement de la réputation de services exposés au grand public par l'altération du bon fonctionnement des systèmes d'IA générative (ex. : *Chatbot*) ;
- exfiltration de données sensibles depuis des systèmes d'IA générative ;
- vol des paramètres (poids) de modèles d'IA propriétaires [11] ;
- latéralisation d'une attaque vers d'autres applications métier interconnectées aux systèmes d'IA générative (ex. : messagerie interne) ;
- sabotage d'applications métier par injection de vulnérabilités dans du code source généré par IA.

La fuite de données sensibles de l'entité doit dans tous les cas être une menace à prendre en compte, quel que soit le cas d'usage du système d'IA générative. Le système d'IA doit intégrer la question des droits d'accès et du respect du besoin d'en connaître dans les réponses qu'il apporte.

Il faut également porter une attention particulière aux scénarios d'attaque indirects impliquant un système d'IA, comme par exemple la génération automatique de contenu pour la mise à disposition d'informations (insertion d'URL malveillante).

Enfin, l'analyse de risque doit tenir compte du scénario de partage de responsabilités retenu pour le projet (cf. chapitre 2). Par exemple, l'utilisation d'un modèle entraîné par un tiers peut engendrer un risque d'attaque par *supply-chain*. Le prestataire tiers, qui n'est pas de confiance, peut entraîner le modèle pour que celui-ci réagisse différemment par rapport au comportement attendu lorsqu'une certaine requête lui est fournie. Une des mesures de réduction du risque pourrait être d'auditer le modèle en question ou bien de ne pas recourir à celui-ci pour des applications critiques.

Une analyse de risque, menée par exemple avec la méthode EBIOS-RM [19], doit ainsi être réalisée le plus tôt possible, c'est-à-dire en amont de la phase d'entraînement.

R2

Mener une analyse de risque sur les systèmes d'IA avant la phase d'entraînement

L'analyse de risque d'un système d'IA doit intégrer les problématiques suivantes :

- cartographier l'ensemble des éléments en lien avec le modèle d'IA : bibliothèques tierces, sources de données, applications interconnectées, etc. ;
- identifier les sous-parties du système d'IA qui traiteront les données de l'organisation, en particulier celles contenues dans les requêtes des utilisateurs ;
- prendre en compte le scénario de partage de responsabilités et la question de la sous-traitance pour chacune des phases ;
- identifier les impacts directs et indirects en cas de réponses erronées ou malveillantes du modèle d'IA aux utilisateurs ;
- considérer la protection des données d'entraînement du modèle d'IA.

Les recommandations du prochain chapitre 5 visent à répondre à ces menaces spécifiques.

11. La connaissance des poids du modèle peut également permettre aux attaquants d'améliorer la capacité de certaines attaques.

5
Recommandations

5.1 Recommandations générales

L'utilisation de bibliothèques et de modules externes non maîtrisés doit être étudiée dès la conception du projet, afin d'identifier les potentielles vulnérabilités liées à ces modules. L'objectif est de se prémunir au maximum d'une attaque dite *supply-chain-attack* visant des composants nécessaires au bon fonctionnement du système d'IA. Il est possible de se reporter au guide de l'ANSSI sur le risque numérique [21] ou encore à la documentation de la CISA à ce sujet [8].

R3

Évaluer le niveau de confiance des bibliothèques et modules externes utilisés dans le système d'IA

Il est recommandé de cartographier l'ensemble des bibliothèques et modules externes utilisés dans le cadre du projet et d'évaluer leur niveau de confiance.

Au même titre que pour les composants logiciels (bibliothèques et modules externes), il est également primordial d'évaluer les sources de données non maîtrisées par l'entité. Ces sources peuvent être des jeux de données d'entraînement récupérés sur Internet, des jeux de validation de la performance d'un modèle ou encore des jeux de données additionnelles utilisés lors de la phase de production.

R4

Évaluer le niveau de confiance des sources de données externes utilisées dans le système d'IA

Il est recommandé de cartographier l'ensemble des sources de données externes utilisées dans le cadre du projet et d'évaluer leur niveau de confiance [12].

De manière générale, il est recommandé d'appliquer des bonnes pratiques de développement sécurisé lors de la conception et de l'implémentation du système d'IA. Ces bonnes pratiques sont parfois regroupées sous le vocable de *DevSecOps* ou de l'expression *security by design*. Pour aller plus loin, il est possible de se reporter au guide de l'ANSSI [28] sur ce thème, ou bien à la documentation du NIST à ce sujet [10], ou encore suivre les préconisations du NCSC-UK [5] et de la CISA [11].

12. Il est possible de s'appuyer sur les critères de la CNIL (https://www.cnil.fr/fr/tenir-compte-de-la-protection-des-donnees-dans-la-collecte-et-la-gestion-des-donnees) ou sur la proposition de *Datasheets for Datasets* (https://arxiv.org/pdf/1803.09010.pdf) pour évaluer un jeu de données externe.

R5

Appliquer les principes de DevSecOps sur l'ensemble des phases du projet

Il est recommandé d'appliquer les bonnes pratiques de développement sécurisé sur l'ensemble des phases du projet, par exemple :

- déployer et sécuriser des chaînes d'intégration et de déploiement en continu (CI/CD) en appliquant le principe de moindre privilège pour l'accès aux outils de ces chaînes CI/CD ;
- mettre en œuvre une gestion sécurisée des secrets utilisés dans toutes les phases du projet ;
- prévoir des tests de sécurité automatisés sur le code source (analyse statique de code) et lors de l'exécution de ce code source (analyse dynamique de code) ;
- protéger en intégrité le code source et sécuriser l'accès à celui-ci (authentification multifacteur, signature du code, droits d'accès, etc.) ;
- recourir à des langages de développement sécurisés (scripts de *fine tuning*, développement du modèle, maintenance, déploiement, etc.).

Afin de pouvoir implémenter un modèle d'IA, il faut stocker les différents paramètres de ce modèle (poids, biais, etc.) dans des fichiers. Plusieurs formats sont disponibles à cet effet, dont certains peuvent présenter un risque d'exécution de code arbitraire, comme par exemple ceux implémentant des fonctionnalités de chargement d'objets sérialisés. Il est donc préférable d'utiliser des formats qui séparent strictement les données de paramètres du modèle et les données de code exécutable de celui-ci.

R6

Utiliser des formats de modèles d'IA sécurisés

Il est recommandé d'utiliser des formats à l'état de l'art du point de vue de la sécurité, comme le format *safetensor* par exemple. Certains formats peu sécurisés, comme le format *pickle*, sont à proscrire.

Les modèles d'IA générative sont amenés à manipuler des données tout au long de leur cycle de vie. Il peut être compliqué d'appliquer des mesures de protection en confidentialité sur ces données, pour les raisons suivantes :

- elles peuvent être issues de sources multiples et sont parfois « mélangées » dans un même jeu : données publiques, données partenaires, données internes, données personnelles, etc.
- leur volume peut être très important, notamment dans le cas de l'entraînement des LLM, ce qui ne facilite pas leur traitement ;
- il peut être nécessaire de les mettre à jour régulièrement, notamment lorsque l'on souhaite ré-entraîner le modèle ;
- il peut être nécessaire de leur appliquer un pré-traitement pour abaisser leur niveau de confidentialité (anonymisation, suppression de certains champs, etc.) ;
- elles peuvent être utilisées lors de différentes phases du projet : pendant la phase d'entraînement du modèle mais également en production en cas d'accès à des données additionnelles par le modèle ;

- elles peuvent inclure les données d'usage du système d'IA pendant la phase de production : données issues des requêtes des utilisateurs et réponses apportées par le modèle d'IA à ces utilisateurs.

Il est important de bien saisir qu'un modèle d'IA hérite de la sensibilité des données qui ont contribué à l'entraîner et également des données qui servent à le ré-entraîner. Un modèle d'IA peut en effet être vulnérable à un phénomène dit de « régurgitation ». Dans certains cas, celui-ci peut par exemple générer des réponses proches des données d'entraînement, révélant ainsi des informations potentiellement sensibles.

En fonction du scénario de partage de responsabilités retenu (cf. chapitre 2), les enjeux de confidentialité des données ne seront pas les mêmes, et les mesures techniques permettant de se prémunir d'une exfiltration de données seront à adapter. Par exemple, dans le cas où l'entité souhaite sous-traiter la phase d'entraînement à un prestataire, il est important de s'assurer que les données stockées et traitées chez ce prestataire sont suffisamment protégées en confidentialité (chiffrement à l'état de l'art, cloisonnement des ressources vis-à-vis d'autres clients, sécurisation des clés utilisées, effacement sécurisé après ré-allocation de ressources, etc.).

Dans le même ordre d'idée, un modèle d'IA propriétaire que l'on souhaite protéger en confidentialité doit faire l'objet de mesures spécifiques de sécurité dans le cas où ce dernier serait stocké dans un environnement qui n'est pas de confiance (ex. : chez un prestataire *Cloud* ou bien embarqué dans un équipement physiquement exposé comme dans le cas de l'IoT).

R7

Prendre en compte les enjeux de confidentialité des données dès la conception du système d'IA

L'étude du projet doit cartographier l'ensemble des jeux de données utilisés à chaque phase du système d'IA : entraînement (jeux de données d'entraînement), déploiement (jeux de tests) et production (données additionnelles, base de données vectorielle, etc.).
Cette étude doit inclure les données d'usage du système d'IA en production, à savoir les requêtes des utilisateurs ainsi que les réponses apportées par le modèle d'IA.
L'analyse peut également traiter le cas de la protection en confidentialité des paramètres du modèle lui-même, par exemple pour des modèles propriétaires.

L'accès à un système d'IA complexifie également l'application du besoin d'en connaître des utilisateurs. Il faut distinguer ici plusieurs catégories de données :

- **Les données d'entraînement** : la gestion de droits d'accès pour les utilisateurs n'est pas possible sur ces données du fait du design des réseaux de neurones ;
- **Les données additionnelles en production** : la gestion de droits d'accès est possible mais elle est dépendante des possibilités offertes par les outils utilisés (RBAC [13]) pour stocker les informations (services de gestion de documents internes, base de données vectorielle, etc.) ;
- **Les données d'usage** : les requêtes des utilisateurs ainsi que les réponses peuvent contenir des données sensibles. Elles sont temporairement stockées lors du traitement dans le système d'IA et parfois utilisées pour ré-entraîner le modèle (ex. : alignement avec RLHF - *Reinforcement learning from human feedback*).

13. *Role Based Access Control*

La question du besoin d'en connaître doit ainsi se reposer à chaque ré-entraînement du modèle, y compris sur des données issues de l'usage du modèle en production (données additionnelles métier, requêtes des utilisateurs, etc.).

R8

Prendre en compte la problématique de besoin d'en connaître dès la conception du système d'IA

Il est important de définir en amont du projet les options structurantes du modèle pour gérer le besoin d'en connaître :

- le choix des données utilisées pour l'entraînement (sans la possibilité de gérer des droits d'accès) et des données additionnelles en production (avec la possibilité de gérer des rôles et des droits d'accès) ;
- la stratégie d'apprentissage du modèle, c'est-à-dire à quel moment est-ce que l'on ré-entraîne le modèle et sur la base de quelles données (données additionnelles métier, requêtes des utilisateurs, réponses du modèle, etc.).

Les IA génératives LLM ont pour la plupart un comportement non-déterministe, et elles peuvent également être sujet à des *hallucinations* [14]. Cette incertitude sur la réponse obtenue pour une requête donnée implique une plus grande vigilance sur les conséquences indirectes de ces réponses. Ainsi, les interactions d'un système d'IA avec d'autres ressources du SI doivent proscrire l'exécution d'actions automatisées critiques pour l'organisation.

Ce principe de précaution doit prévaloir et un système d'IA générative ne doit pas pouvoir prendre des décisions critiques ayant un impact fort sur le métier ou la protection des biens et des personnes, sans un contrôle humain (ex. : validation dans une IHM). Dans ces cas particuliers, la capacité de discernement humaine permet de réduire le risque de scénarios qui peuvent représenter un danger pour l'organisation.

Il est par exemple important de ne pas utiliser un système d'IA pour automatiser des actions d'administration critiques sur l'infrastructure technique de l'entité (ex. : découverte et déploiement automatique de configurations réseaux ou de règles de pare-feux).

R9

Proscrire l'usage automatisé de systèmes d'IA pour des actions critiques sur le SI

Un système d'IA doit être configuré de manière à ne pas être en mesure d'exécuter de manière automatisée des actions critiques sur le SI.
Ces actions peuvent être des actions critiques d'un point de vue métier (transactions bancaires, production de contenu public, impact direct sur des personnes, etc.) ou bien des actions critiques sur l'infrastructure du SI (reconfiguration de composants réseaux, créations d'utilisateurs à privilège, déploiement de serveurs virtuels, etc.).

Les rôles et les droits d'accès des développeurs et des administrateurs du système d'IA doivent être strictement définis et appliqués dès le début du projet. Les principes d'administration sécurisée, tels

14. Phénomène dans lequel un modèle génère du contenu erroné qui n'est pas basé sur des données réelles.

que décrits dans le guide de l'ANSSI à ce sujet [26] doivent être appliqués dans toutes les phases du cycle de vie du système d'IA.

R10

Maîtriser et sécuriser les accès à privilèges des développeurs et des administrateurs sur le système d'IA

L'ensemble des opérations à privilèges sur le système d'IA doit respecter les bonnes pratiques d'administration sécurisée, notamment :

- les opérations à privilège doivent être définies et leur déclenchement doit être validé : ré-entraînement, modification des jeux de données, nouvelle interconnexion avec une application, changement d'hébergement, etc. ;
- les opérations à privilège doivent être réalisées avec des comptes dédiés et depuis un poste d'administration dédié à cet usage ;
- le principe de moindre privilège doit être appliqué et l'usage de jetons d'authentification (*token*) temporaires doit être privilégié ;
- l'environnement de développement doit être maîtrisé et administré au même niveau de sécurité que l'environnement de production.

L'hébergement du système d'IA, quelle que soit sa phase, doit être étudié. Le niveau de sécurité doit être cohérent avec les besoins de sécurité du projet, et notamment les besoins en confidentialité des données utilisées dans chacune des phases.

Ce point doit notamment être strictement respecté pour la phase d'entraînement du modèle car des menaces importantes existent lors de cette phase, comme cela a été vu précédemment dans le chapitre 4.

Attention

Les modèles d'IA sont considérés du même niveau de sensibilité que les données qui ont servi à sa conception et à son entraînement. La règle R9 [12] de la circulaire « Cloud au centre » [13] doit s'appliquer pour le cas de l'administration publique.

R11

Héberger le système d'IA dans des environnements de confiance cohérents avec les besoins de sécurité

L'hébergement du système d'IA lors des 3 phases du cycle de vie doit être cohérente avec les besoins de sécurité du projet, et notamment les besoins en confidentialité et en intégrité. En particulier, la sécurisation des données d'entraînement du modèle (au repos, en transit, lors d'un traitement) ne doit pas être négligée.

Les 3 environnements d'entraînement, de déploiement et de production d'un système d'IA doivent être cloisonnés entre eux. Cette mesure permet de se prémunir d'un risque de latéralisation entre les environnements. Cela est d'autant plus important que les populations ayant accès à chaque environnement ne sont généralement pas les mêmes.

R12

Cloisonner chaque phase du système d'IA dans un environnement dédié

Il est recommandé de cloisonner les 3 environnements techniques correspondant à chacune des phases du cycle de vie du système d'IA. Ce cloisonnement peut porter sur :

- un cloisonnement réseau : chaque environnement est intégré dans un réseau physiquement ou logiquement dédié ;
- un cloisonnement système : chaque environnement dispose de ses propres serveurs physiques ou hyperviseurs dédiés ;
- un cloisonnement du stockage : chaque environnement dispose de son propre matériel de stockage ou de disques dédiés. Au minimum, un cloisonnement logique est appliqué ;
- un cloisonnement des comptes et des secrets : chaque environnement dispose de ses propres comptes utilisateurs et administrateurs et de secrets distincts.

Dans le cas d'un système d'IA exposé sur Internet, il est recommandé de suivre les préconisations de l'ANSSI pour la conception d'une passerelle Internet sécurisée [24].

R13

Implémenter une passerelle Internet sécurisée dans le cas d'un système d'IA exposé sur Internet

Dans le cas d'un système d'IA exposé sur Internet, il est recommandé de suivre les bonnes pratiques de cloisonnement du guide de l'ANSSI à ce sujet, notamment :

- dédier une fonction de *reverse-proxy* avant l'accès au service web du système d'IA ;
- mettre en place deux zones logiques pour le filtrage réseau à l'aide de pare-feux : un filtrage externe en frontal d'Internet et un filtrage interne avant l'accès au système d'IA ;
- ne pas exposer un annuaire interne de l'entité pour l'authentification sur le système d'IA ;
- éviter de mutualiser sur un même hyperviseur des fonctions de sécurité distinctes de la passerelle Internet sécurisée (pare-feux, *reverse-proxy*, serveur de journalisation, etc.).

Si l'entité fait le choix d'un *Cloud* public [15] pour exposer son service, il est recommandé de choisir un prestataire qualifié SecNumCloud [32] si les besoins de sécurité l'exigent.

R14

Privilégier un hébergement SecNumCloud dans le cas d'un déploiement d'un système d'IA dans un Cloud public

Si l'entité fait le choix d'utiliser un hébergement dans un *Cloud* public, il est recommandé de privilégier une offre de confiance SecNumCloud dans les cas suivants :

- les données traitées par le système d'IA sont considérées comme sensibles ;

15. Un *Cloud* public désigne un service d'hébergement mutualisé entre plusieurs clients et exposé sur Internet.

- l'impact du système d'IA sur le métier est considéré comme critique ;
- les utilisateurs du système d'IA ne sont pas considérés comme de confiance.

Lors de la conception du projet, il faut prévoir systématiquement un mode dégradé sans IA pour répondre aux besoins métier, en cas d'indisponibilité ou de défaillance du système nominal.

R15

Prévoir un mode dégradé des services métier sans système d'IA

Afin de prévenir des dysfonctionnements ou des incohérences dans les réponses apportées par le modèle d'IA, il est recommandé de prévoir au minimum une procédure de contournement du système d'IA pour les utilisateurs, afin de répondre aux besoins métier.

Le déploiement de systèmes d'IA générative et de LLM implique généralement l'usage de GPU [16] dans un objectif de performances du système, que ce soit dans la phase d'entraînement ou de production.

Ces GPU peuvent potentiellement traiter des données sensibles en lien avec les opérations du modèle d'IA. Dans un objectif de se protéger d'une fuite de données, il est recommandé de dédier ces composants matériels GPU au système d'IA et de ne pas les mutualiser avec d'autres applications métier du SI. Les GPU peuvent en revanche être mutualisés entre plusieurs modèles d'IA, sous réserve que ces derniers correspondent à un même niveau de sensibilité et à des besoins de sécurité homogènes.

R16

Dédier les composants GPU au système d'IA

Il est recommandé de dédier les composants physiques GPU aux traitements réalisés par le système d'IA. Dans le cas de la virtualisation, il est recommandé que les hyperviseurs ayant accès aux cartes GPU soient dédiés au système d'IA, ou bien au minimum qu'il y ait une fonction de filtrage matériel (ex. : IOMMU [17]) permettant de restreindre les accès des machines virtuelles à la mémoire de ces cartes GPU.

Comme la plupart des applications métier, les systèmes d'IA peuvent être soumis à des attaques par canaux auxiliaires. Ces attaques peuvent avoir pour objectifs d'exfiltrer des informations sensibles ou de perturber le bon fonctionnement des systèmes d'IA. Si la plupart de ces attaques ne sont pas propres à un système d'IA, certaines peuvent néanmoins s'appuyer sur des mécanismes spécifiques aux systèmes d'IA génératives [18].

R17

Prendre en compte les attaques par canaux auxiliaires sur le système d'IA

Il est recommandé de s'assurer que le système d'IA n'est pas vulnérable à des attaques par canaux auxiliaires (temporels, consommation, etc.) qui pourraient par exemple permettre à un attaquant de reconstruire une réponse apportée par un modèle d'IA.

16. *Graphics processing unit*
17. *Input–output memory management unit*
18. cf. par exemple `https://cdn.arstechnica.net/wp-content/uploads/2024/03/LLM-Side-Channel.pdf`

5.2 Recommandations pour la phase d'entraînement

La question de la confidentialité des données a fait l'objet d'une recommandation générale précédemment (cf. R7). En particulier, et au vu des nombreuses vulnérabilités publiées sur les outils d'IA générative, il est préférable de partir du postulat qu'un utilisateur ayant accès à un modèle d'IA entraîné pourrait potentiellement avoir accès aux données d'entraînement de ce même modèle.

Pour réduire les risques liés à la confidentialité des données d'entraînement, il est parfois envisagé de recourir à un processus d'anonymisation ou bien de générer un jeu de données synthétiques à partir des données brutes d'origine. Ces mesures peuvent répondre dans certains cas aux enjeux de protection de l'information, mais il convient néanmoins d'être vigilant sur l'existence d'attaques visant à retrouver l'information initiale à partir de données anonymisées ou synthétiques [19] : attaques par inférence d'attribut ou d'appartenance, ré-identification à partir de croisements avec d'autres jeux de données, etc.

R18

Entraîner un modèle d'IA uniquement avec des données légitimement accessibles par les utilisateurs

Il est fortement recommandé d'entraîner un modèle avec des données dont la sensibilité est cohérente avec le besoin d'en connaître des utilisateurs.

Comme cela a été vu précédemment, des attaques ciblant spécifiquement la phase d'entraînement d'un modèle sont possibles, comme par exemple l'injection de données malveillantes dans les jeux de données d'entraînement, ou encore la modification de certaines données pour générer un dysfonctionnement du modèle, une fois celui-ci déployé en production.

R19

Protéger en intégrité les données d'entraînement du modèle d'IA

Il est recommandé de s'assurer de l'intégrité des données d'entraînement du modèle tout au long du cycle d'entraînement. Cette protection peut prendre la forme d'une vérification systématique de la signature ou de l'empreinte (*hash*) des fichiers utilisés ou bien des archives compressés de l'ensemble de ces données.

R20

Protéger en intégrité les fichiers du système d'IA

Il est recommandé de protéger en intégrité les fichiers du modèle entraîné, et de contrôler régulièrement que ceux-ci n'ont pas été altérés. La recommandation vaut également par extension pour tous les fichiers inhérents au fonctionnement du système d'IA (scripts, binaires, etc.).

Dans la majorité des cas d'usage, un modèle d'IA entraîné ne doit pas faire l'objet d'une modification ou d'un ajustement constant de ses paramètres. Dans le cas où l'on constate un dysfonctionnement, ou bien lorsque l'on cherche à optimiser les performances du modèle, il est préférable d'exécuter les opérations de ré-entraînement de ce dernier en utilisant l'environnement d'entraînement dédié à cet usage.

19. Il est possible de se référer aux travaux de la CNIL à ce sujet : https://linc.cnil.fr/donnees-synthetiques-et-lhomme-crea-les-donnees-son-image-22.

A ce titre, les méthodes d'apprentissage en continu ou aussi appelées apprentissage en ligne (le modèle apprend en temps-réel à partir des données envoyées en entrée) sont à éviter au possible. En effet, l'utilisation de méthodes d'apprentissage hors ligne, à partir de jeux de données sélectionnés et testés, réduit les risques de dysfonctionnement du modèle ou d'empoisonnement de celui-ci.

Le ré-entraînement d'un modèle d'IA peut être réalisé de manière récurrente et fixe (ex. : tous les mois), être déclenché lorsqu'un écart de performance franchit un seuil donné ou lorsque les données d'entraînement ne sont plus pertinentes, ou encore à la demande de manière ponctuelle.

R21

Proscrire le ré-entraînement du modèle d'IA en production

Il est fortement recommandé de ne pas ré-entraîner un modèle d'IA directement en production. Cette action de ré-entraînement doit démarrer avec le cycle en 3 phases, dans les environnements adéquats pour chacune des phases.

5.3 Recommandations pour la phase de déploiement

Le déploiement d'un système d'IA générative doit s'appuyer sur un environnement de déploiement sécurisé, reposant par exemple sur des chaînes CI/CD maîtrisées et durcies.

Ces chaînes CI/CD doivent être opérées depuis un SI d'administration et depuis des postes d'administrateurs dédiées et durcis.

R22

Sécuriser la chaîne de déploiement en production des systèmes d'IA

Il est recommandé d'opérer le déploiement des systèmes d'IA générative depuis un SI d'administration, en respectant les bonnes pratiques du guide d'administration sécurisée [26] de l'ANSSI.

Il faut prévoir une phase d'audit de sécurité par des équipes spécialisées et formées aux spécificités des systèmes d'IA. Cette phase doit avoir lieu avant le déploiement en production afin de tester les vulnérabilités inhérentes aux systèmes d'IA (attaques adverses, etc.).

R23

Prévoir des audits de sécurité des systèmes d'IA avant déploiement en production

Il est recommandé de prévoir des tests de robustesse et de sécurité des systèmes d'IA. Ces tests peuvent être :

- des tests d'intrusion standards sur les composants techniques usuels d'un système d'IA : serveurs web, orchestrateur, base de données, etc.
- des tests de sécurité sur les développements réalisés dans le système d'IA (via des outils SAST ou DAST par exemple) ;
- des tests automatisés [20] visant spécifiquement des vulnérabilités liées aux modèles d'IA (attaques adverses, extraction du modèle, etc.) ;
- des tests manuels d'auditeurs visant spécifiquement à tester la robustesse d'un modèle d'IA générative sur des scénarios d'attaques plus sophistiqués.

Pour la réalisation d'audits de sécurité sur un système d'IA générative, il est possible de faire appel à des prestataires PASSI [30] qualifiés par l'ANSSI.

R24

Prévoir des tests fonctionnels métier des systèmes d'IA avant déploiement en production

Il est recommandé de prévoir des tests de performance et de qualité des réponses apportées par un système d'IA générative.

Information

Les tests fonctionnels du système d'IA peuvent avoir lieu en continu à une fréquence donnée et pas uniquement lors de déploiements. Cela peut permettre de détecter en avance de phase un dysfonctionnement du modèle et ainsi corriger de manière plus réactive.

5.4 Recommandations pour la phase de production

Comme indiqué précédemment, il est difficile d'appliquer le principe de besoin d'en connaître sur les données d'entraînement d'un modèle, celui-ci pouvant subir des attaques qui visent à extraire ces données par interrogation du modèle (régurgitation).

De même, certaines requêtes malveillantes peuvent avoir comme objectif le détournement du service d'IA générative, par exemple en provoquant des hallucinations ou des réponses erronées.

Dans une démarche de défense en profondeur, il est recommandé d'étudier la possibilité de détecter ou de bloquer certaines requêtes malveillantes ayant par exemple pour objectif d'extraire des informations du modèle ou des données additionnelles (ces données additionnelles pouvant inclure les entrées des utilisateurs dans certains cas d'usage).

Cette protection peut également être pertinente pour se prémunir du risque de fuite du modèle. En effet, si le modèle a été entraîné sur des données sensibles, la fuite des paramètres du modèle peut entraîner la fuite de certaines de ces données par certaines attaques (ex. : attaque par inversion de modèle ou attaque par inférence d'appartenance). À ce titre, les réponses aux utilisateurs doivent être les plus simples possibles (chaîne de caractères exclusivement) et ne doivent pas retourner de vecteur de score ayant servi à la prédiction, ni tout autre mécanisme interne au modèle.

Enfin, il peut être pertinent, selon les cas d'usage, de définir une limite à la taille des réponses apportées par le modèle d'IA. Cela peut ainsi réduire le risque de fuite de données par régurgitation.

R25

Protéger le système d'IA en filtrant les entrées et les sorties des utilisateurs

Il est recommandé de mettre en place des fonctions permettant de se prémunir d'une fuite de données ou d'une fuite de modèle dans les réponses :

- une fonction de filtre de requêtes malveillantes des utilisateurs avant envoi au

20. À titre d'exemples, plusieurs outils spécialisés existent comme https://github.com/microsoft/responsible-ai-toolbox, https://github.com/Trusted-AI/adversarial-robustness-toolbox, ou encore https://github.com/protectai/ai-exploits.

modèle ;

- une fonction de filtre de requêtes jugées non légitimes d'un point de vue métier ;
- une fonction de filtre d'informations internes du modèle (paramètres, entraînement) dans les réponses ;
- une fonction de filtre d'informations définies comme sensibles dans les réponses (ex. : coordonnées personnelles, références de projets, etc.) ;
- une limite sur la taille des réponses (nombre de caractères maximum).

Les interactions du système d'IA avec d'autres applications métier ou d'autres ressources techniques du SI peuvent être une source de vulnérabilités.

Ces interactions prennent souvent la forme de *plugins* proposés par les éditeurs de modèles d'IA. Ces *plugins* vont ainsi permettre d'interconnecter le système d'IA avec des outils bureautiques, des réseaux sociaux, ou encore des composants d'infrastructure potentiellement critiques (gestionnaire d'identités, ressources réseaux, etc.).

Ces interactions peuvent également faciliter la latéralisation d'un attaquant sur le SI, si celui-ci profite d'une vulnérabilité sur le système d'IA.

La littérature évoque très souvent les risques d'injection de requête indirecte (*indirect prompt injection*) et les problèmes qui peuvent résulter de l'envoi de données non-maîtrisées à un modèle d'IA générative [21] (par exemple le contenu d'un mail reçu ou d'une page Web issue d'une recherche). Ces usages sont d'autant plus problématiques dans le cas où des actions pourraient être effectuées sans une validation humaine (cf. recommandation R9).

Il est donc primordial d'avoir une maîtrise des interactions du système d'IA avec d'autres ressources du SI.

R26

Maîtriser et sécuriser les interactions du système d'IA avec d'autres applications métier

L'ensemble des interactions et des flux réseaux du système d'IA doit être documenté et validé. Les flux réseaux entre le système d'IA et d'autres ressources doivent respecter l'état de l'art en matière de sécurité :

- ils doivent être strictement filtrés au niveau réseau, chiffrés et authentifiés (ex. : en suivant le guide TLS de l'ANSSI [22]) ;
- ils doivent utiliser des protocoles sécurisés (ex. : *OpenID Connect*) en cas d'usage d'un fournisseur d'identité [23] ;
- ils doivent intégrer un contrôle des autorisations d'accès à la ressource en complément de l'authentification ;
- ils doivent faire l'objet d'une journalisation au niveau de granularité adéquat.

21. Il est possible de se référer à l'article « *Not what you've signed up for : Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection* » pour plus de détails : `https://arxiv.org/abs/2302.12173`

R27

Limiter les actions automatiques depuis un système d'IA traitant des entrées non-maîtrisées

Il est fortement recommandé de limiter voire de proscrire les actions automatiques sur le SI déclenchées depuis un système d'IA et à partir d'entrées non-maîtrisées (ex. : données issues d'Internet ou de mails, etc.).

En fonction du cas d'usage du système d'IA et de sa criticité d'un point de vue métier, il peut être opportun de le déployer dans un ou plusieurs environnements dédiés, c'est-à-dire non mutualisés avec d'autres applications métier de l'entité.

R28

Cloisonner le système d'IA dans un ou plusieurs environnements techniques dédiés

Il est recommandé que le système d'IA soit cloisonné dans des zones logiques dédiées, afin de limiter les risques de latéralisation d'un attaquant qui aurait compromis ce système.

La journalisation des actions sur le système d'IA doit être réalisée avec une granularité d'information adéquate, en particulier sur les entrées et les sorties du modèle d'IA.

Dans un objectif de traçabilité et de compréhension du système d'IA, il est important de bien distinguer les requêtes réalisées par les utilisateurs et les données réellement envoyées au modèle d'IA. En effet, pour des raisons de performance mais aussi de sécurité, les requêtes des utilisateurs peuvent faire l'objet d'un pré-traitement et d'une mise en forme spécifique avant l'envoi au modèle.

Ces deux informations sont cruciales pour faciliter la gestion d'un incident et doivent pouvoir être tracées dans des journaux applicatifs du système d'IA. L'objectif est de pouvoir reconstituer entièrement un événement sur le système d'IA, en cas de requête malveillante par exemple.

En ce qui concerne l'architecture d'un système de journalisation, il est possible de s'appuyer sur le guide généraliste de l'ANSSI à ce sujet [27].

R29

Journaliser l'ensemble des traitements réalisés au sein du système d'IA

Il est recommandé de journaliser au bon niveau de granularité l'ensemble des traitements réalisés sur le système d'IA, notamment :

- les requêtes des utilisateurs (en prenant garde à leur protection si ces requêtes contiennent des données sensibles) ;
- les traitements réalisés en entrée sur cette requête avant l'envoi au modèle ;
- les appels à des *plugins* ;
- les appels à des données additionnelles ;
- les traitements réalisés par les filtres en sortie ;
- les réponses aux utilisateurs.

Attention

La journalisation des données des utilisateurs doit respecter les exigences de la CNIL [9] concernant la protection des données personnelles (comme prévu par le RGPD) et notamment pour la durée de conservation de ces données sur le système d'IA.

5.5 Cas particulier de la génération de code source assistée par l'IA

Les outils d'IA générative peuvent être spécialisés et spécifiquement entraînés pour générer du code source dans plusieurs langages de programmation.

Ces moyens peuvent permettre aux développeurs un gain de temps mais comportent aussi des risques sur la qualité du code (introduction de vulnérabilités) ou d'insertion de porte dérobée dans le cas où un attaquant aurait compromis le modèle.

Il est donc important de faire preuve de vigilance sur le code source généré par IA.

R30

Contrôler systématiquement le code source généré par IA

Le code source généré par IA doit faire l'objet de mesures de sécurité afin de vérifier son innocuité :

- proscrire l'exécution automatique de code source généré par IA dans l'environnement de développement ;
- proscrire le *commit* automatique de code source généré par IA dans les dépôts ;
- intégrer un outil d'assainissement de code source [3, 4] généré par IA dans l'environnement de développement ;
- vérifier l'innocuité des bibliothèques référencées dans le résultat du code source généré par IA ;
- faire contrôler régulièrement par un humain la qualité du code source généré à partir de requêtes types suffisamment sophistiquées.

R31

Limiter la génération de code source par IA pour des modules critiques d'applications

Il est fortement recommandé de ne pas utiliser un outil d'IA générative pour générer des blocs [22] de code source destinés à des modules critiques d'applications :

- les modules de cryptographie (authentification, chiffrement, signature, etc.) ;
- les modules de gestion des droits d'accès des utilisateurs et administrateurs ;
- les modules de traitement de données sensibles.

R32

Sensibiliser les développeurs sur les risques liés au code source généré par IA

Il est recommandé d'effectuer des campagnes de sensibilisation sur les risques liés à l'utilisation de code source généré par IA. Cette sensibilisation peut s'appuyer sur des rapports publics sur ce sujet ou bien des papiers de recherche [23] démontrant la présence de vulnérabilités dans le code généré par IA.
En complément, les développeurs peuvent également être formés sur les outils d'IA pour l'optimisation de leurs requêtes (*prompt engineering* [24]) afin d'améliorer la qualité et la sécurité du code généré.

Information

Selon les cas d'usages, il peut également être pertinent d'entraîner spécifiquement un modèle (étape d'alignement) pour qu'il ne puisse pas être en mesure de générer du code volontairement malveillant.

5.6 Cas particulier de services d'IA grand public exposés sur Internet

Dans le cas où l'entité souhaite proposer au grand public un service reposant sur de l'IA générative, il convient d'apporter une vigilance particulière à la sécurisation de ce service, à cause de sa forte exposition.

L'atteinte à l'image ou à la réputation de l'entité peut être une menace supplémentaire à identifier lors de l'analyse de risque.

R33

Durcir les mesures de sécurité pour des services d'IA grand public exposés sur Internet

Il est recommandé d'apporter une attention particulière sur certaines mesures de sécurité pour les services exposés au grand public, notamment :

- entraîner le modèle d'IA uniquement à partir de données publiques ;
- s'assurer que les utilisateurs du système d'IA ont fait l'objet d'une authentification au préalable ;
- analyser systématiquement les requêtes des utilisateurs sur le système d'IA ;
- faire un contrôle et une validation des réponses avant l'envoi aux utilisateurs ;
- protéger en confidentialité les données des utilisateurs (historique des requêtes et des réponses, etc.) ;

22. Un bloc désigne ici un ensemble complet d'instructions du code source, par exemple la définition complète d'une fonction, d'une procédure, d'une classe d'objets, d'un script shell, etc.
23. Les rapports de *Snyk* (`https://snyk.io/fr/reports/ai-code-security/`) peuvent éventuellement être cités à titre d'exemple ou bien les recherches menées par l'université de Stanford à ce sujet (`https://arxiv.org/pdf/2211.03622.pdf`).
24. Il est par exemple possible de combiner une première requête de génération de code par IA, suivie d'un test d'analyse statique de ce code, puis enfin une seconde requête demandant à l'IA de corriger les vulnérabilités détectées dans le code qui avait été généré.

- mettre en place des mesures contre les dénis de service distribués (DDoS) [18];
- sécuriser le service web en frontal des utilisateurs [25].

5.7 Cas particulier de l'utilisation de solutions d'IA générative tierces

Dans cette dernière section, le guide traite le cas particulier où l'entité n'est pas en situation de gestionnaire et de maîtrise d'un service d'IA générative, mais est cliente d'un service tiers d'IA générative (cf. scénarios de partage de responsabilités du chapitre 2). L'objet de cette dernière section est de faire un rappel des points de vigilance à prendre en compte pour les utilisateurs de ces services tiers.

De par leur facilité d'usage, il est tentant de recourir à des outils d'IA générative disponibles sur Internet pour traiter des données métier, par exemple pour la traduction de textes. Le fait d'envoyer des informations (texte, images, documents) à un service d'IA générative grand public revient à déposer ces mêmes informations sur un espace de stockage leur appartenant.

Le cloisonnement entre les clients ainsi que la protection en confidentialité des données envoyées au système d'IA sur Internet ne sont pas maîtrisés et reposent uniquement sur la confiance envers le prestataire. À ce titre, il est important de noter que dans la majorité des offres, les données envoyées au service sont collectées et utilisées par le prestataire à des fins d'optimisation des modèles [25].

Il est donc primordial de ne surtout pas envoyer de données sensibles à des services d'IA générative tiers grand public comme par exemples *ChatGPT*, *Gemini*, *Copilot*, *DeepL* (Traduction de texte) ou encore *Perplexity* pour ne citer que les plus populaires. Les données concernées sont notamment :

- des données de niveau *Diffusion Restreinte* [29] ou classifiées de défense [31];
- des travaux de recherche relevant de la PPST [20];
- des données personnelles (vie privée, coordonnées, etc.);
- des données contractuelles, juridiques ou financières de l'entreprise;
- des secrets informatiques, comme des mots de passe ou des jetons d'authentification (clés d'API).

R34

Proscrire l'utilisation d'outils d'IA générative sur Internet pour un usage professionnel impliquant des données sensibles

L'entité cliente n'ayant pas la maîtrise du service d'IA générative, il n'est pas possible de s'assurer que la protection en confidentialité des données soumises en entrée respecte les besoins de sécurité de l'entité.
Par mesure de précaution, il est donc obligatoire de ne jamais intégrer de données sensibles de l'entité dans les requêtes des utilisateurs.

25. cf. la politique d'utilisation de *ChatGPT* par exemple : https://help.openai.com/en/articles/7842364-how-chatgpt-and-our-language-models-are-developed

Attention

Cette recommandation concerne également l'utilisation d'outils d'IA générative dans le but de générer des jeux de données synthétiques pour l'entraînement ou le *fine-tuning* d'un modèle d'IA.

Certains outils tiers d'IA générative peuvent proposer des connexions avec des outils bureautiques ou des applications métier usuelles sur Internet. Une attention particulière doit être portée à la configuration des droits d'accès des outils d'IA générative sur les données métier de l'entité : mails, espace documentaire, dépôts de code source, services d'audioconférence et de visioconférence, etc.

R35

Effectuer une revue régulière de la configuration des droits des outils d'IA générative sur les applications métier

Il est recommandé de faire une revue des droits d'accès des outils d'IA générative dès l'activation du produit dans l'entité, afin de s'assurer que les droits positionnés par défaut ne sont pas trop laxistes ou trop ouverts par conception.
Enfin, une revue régulière des droits d'accès doit être réalisée (ex. : tous les mois), afin notamment de s'assurer que les mises à jour fonctionnelles et de sécurité du produit ne viennent pas impacter le besoin d'en connaître pour les utilisateurs.

Liste des recommandations

Bibliographie

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

[2] *CNIL - Intelligence artificielle (IA)*.
Site institutionnel, CNIL.
https://www.cnil.fr/fr/intelligence-artificielle-ia.

[3] *NIST - Source Code Security Analyzers*.
Site institutionnel, 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*.
Site institutionnel, NCSC-UK, novembre 2018.
https://www.ncsc.gov.uk/collection/developers-collection.

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

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

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

[9] *CNIL - IA : comment être en conformité avec le RGPD ?*
Site institutionnel, CNIL, avril 2022.
https://www.cnil.fr/fr/intelligence-artificielle/ia-comment-etre-en-conformite-avec-le-rgpd.

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

[11] *CISA - Defending Continuous Integration/Continuous Delivery (CI/CD) Environments*.
Site institutionnel, CISA, juin 2023.
https://media.defense.gov/2023/Jun/28/2003249466/-1/-1/0/CSI_DEFENDING_CI_CD_ENVIRONMENTS.PDF.

[12] *DINUM - Le Cloud pour les administrations*.
Site institutionnel, DINUM, Mai 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 centre*.
Site institutionnel, LEGIFRANCE, Mai 2023.
https://www.legifrance.gouv.fr/download/pdf/circ?id=45446.

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

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

[16] *NIST - Adversarial Machine Learning : A Taxonomy and Terminology of Attacks and Mitigations*.
Site institutionnel, NIST, janvier 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, septembre 2017.
https://cyber.gouv.fr/hygiene-informatique.

[18] *Comprendre et anticiper les attaques DDoS*.
Guide Version 1.0, ANSSI, mars 2015.
https://cyber.gouv.fr/guide-ddos.

[19] *La méthode EBIOS Risk Manager - Le Guide*.
Guide ANSSI-PA-048 v1.0, ANSSI, octobre 2018.
https://cyber.gouv.fr/ebios-rm.

[20] *Protection du potentiel scientifique et technique de la nation*.
Guide ANSSI-PA-049 v1.0, ANSSI, avril 2018.
https://cyber.gouv.fr/guide-zrr.

[21] *Maîtrise du risque numérique - l'atout confiance*.
Guide ANSSI-PA-070 v1.0, ANSSI, novembre 2019.
https://cyber.gouv.fr/publications/maitrise-du-risque-numerique-latout-confiance.

[22] *Recommandations de sécurité relatives à TLS*.
Guide ANSSI-PA-035 v1.2, ANSSI, mars 2020.
https://cyber.gouv.fr/guide-tls.

[23] *Recommandations pour la sécurisation de la mise en œuvre du protocole OpenID Connect*.
Guide ANSSI-PA-080 v1.0, ANSSI, septembre 2020.
https://cyber.gouv.fr/guide-oidc.

[24] *Recommandations relatives à l'interconnexion d'un système d'information à Internet*.
Guide ANSSI-PA-066 v3.0, ANSSI, juin 2020.
https://cyber.gouv.fr/guide-interconnexion-si-internet.

[25] *Recommandations pour la mise en œuvre d'un site Web : maîtriser les standards de sécurité côté navigateur*.
Guide ANSSI-PA-009 v2.1, ANSSI, avril 2021.
https://cyber.gouv.fr/guide-sites-web.

[26] *Recommandations relatives à l'administration sécurisée des systèmes d'information*.
Guide ANSSI-PA-022 v3.0, ANSSI, mai 2021.
https://cyber.gouv.fr/guide-admin-si.

[27] *Recommandations de sécurité pour l'architecture d'un système de journalisation*.
Guide DAT-PA-012 v2.0, ANSSI, janvier 2022.
https://cyber.gouv.fr/guide-journalisation.

[28] *Les essentiels - DevSecOps*.
Guide Version 1.0, ANSSI, février 2024.
https://cyber.gouv.fr/publications/devsecops.

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

[30] *Prestataires d'audit de la sécurité des systèmes d'information. Référentiel d'exigences*.
Référentiel Version 2.1, ANSSI, octobre 2015.
https://cyber.gouv.fr/referentiels-dexigences-pour-la-qualification.

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

[32] *Prestataires de services d'informatique en nuage (SecNumCloud). Référentiel d'exigences*.
Référentiel Version 3.2, ANSSI, mars 2022.
https://cyber.gouv.fr/secnumcloud.

Version 1.0 - 29/04/2024 - ANSSI-PA-102
Licence ouverte / Open Licence (Étalab - v2.0)
ISBN : 978-2-11-167156-0 (papier)
ISBN : 978-2-11-167157-7 (numérique)
Dépôt légal : Avril 2024

AGENCE NATIONALE DE LA SÉCURITÉ DES SYSTÈMES D'INFORMATION

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

RÉPUBLIQUE FRANÇAISE
Liberté
Égalité
Fraternité