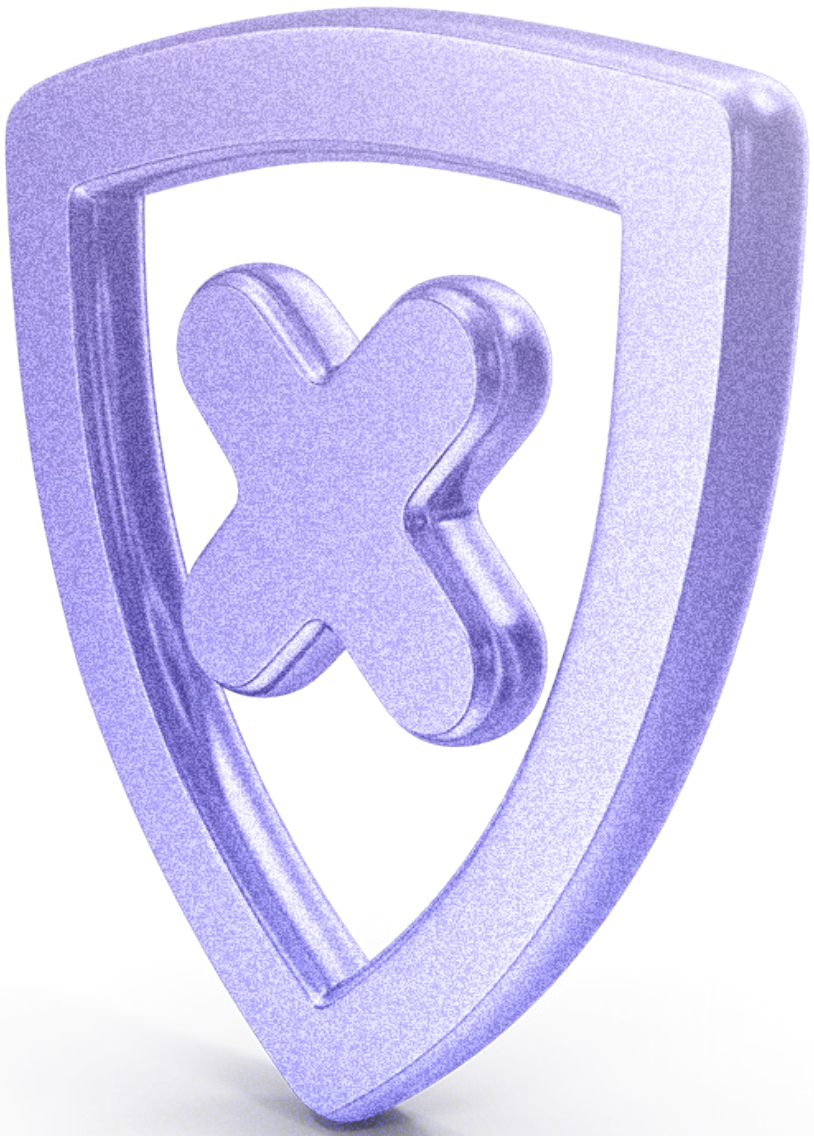
FS-ISAC

Adversarial AI Frameworks: Taxonomy, Threat Landscape, and Control Frameworks



February 2024

Contents

AI RISK WORKING GROUP MISSION STATEMENT

Provide recommendations and best practices to the financial services industry on prescribing pragmatic Generative AI (GenAI) threats, risks, and controls that enable the AI security objectives below.

- Survey and document existing landscape of AI threat, risk, and control frameworks with focus on generative AI.
- Identify gaps and opportunities for enhancements to existing frameworks.
- Influence enhancements to industry frameworks and security tooling.
- Standardize a generative AI threat taxonomy for the financial sector.

EXECUTIVE SUMMARY

The Adversarial AI Frameworks document aims to provide an approach to tracking and assessing AI-enabled threats in the financial services sector, specifically focusing on recent developments in generative AI. To that end, this framework assesses risk levels relating to the integration of GenAI into business operations, adopting a threat-led approach for the implementation of controls and mitigations. This review is applicable to many GenAI modalities, but utilizes large language models (LLMs) as a focus point due to their broader applications and common baseline for multi-modal models.

GenAI is an emerging technology; the evolving nature of the technology limits the ability to provide examples and resources around secure adoption and implementation. Many existing frameworks are unable to adequately convey the breadth of threats posed.

This document derives its conclusions, framework, and taxonomy from a variety of sources, and suggests control implementations agnostic of vendor.

The threats discussed in this document include those related to AI models, but extend to the data such models are trained and tested on, the third-party components, plug-ins, and libraries utilized in their development, as well as the platform such models can be hosted on. The focus is the impact to the AI application, and thus some instances will include discussion of threats with existing mitigations, considerations, and best practices from many organizations. Such instances may bear repeating to sufficiently discuss the threat to or from an AI application, but are often best mitigated by existing cyber hygiene and best practices.

This report does not make recommendations on risk tolerance nor on the level of residual risk, as such assessments are use-case specific and individual to each organization.

This document contains a taxonomy of attacks and defines key terminology used in the discussion of AI threats. The taxonomy and list of technical explanations contextualizes these threats to provide an easily digestible and applicable hierarchy, as well as a financial services sector relevant document that can be of use to non-financial sector entities. This taxonomy incorporates existing literature and public taxonomies to remain consistent with public and adopted alternatives – including the NIST Adversarial Machine Learning: A Taxonomy and Terminology of Attacks and Mitigations[1] – to ensure consistency in literature, reporting, and discussions, as well as underpin further work on this topic. However, the FS-ISAC AI Risk Working Group's framework was designed to comprehensively cover AI threats to financial sector entities while considering additional risks not addressed in currently published alternatives.

Risk Considerations Unique to FS-ISAC's Adversarial AI Frameworks	
Copilot security	FS-ISAC framework highlights AI trust layer deficiencies in copilots.
Hallucinations	FS-ISAC framework considers hallucinations and mitigations.
Architecture-focused with layered defenses	FS-ISAC framework enumerates GenAI assets under threat, architecture.

Comparison of Risk Considerations in FS-ISAC's Adversarial AI Frameworks to NIST's AI Risk Management Framework	
Privacy by Design	FS-ISAC framework considers pragmatic privacy controls (e.g., LINDDUN). NIST approach is academic.
Cryptography, API Security	FS-ISAC framework is all encompassing.
AI Software Supply Chain	FS-ISAC approach considers LLM SBOMs.
Multi-Modal Security	Comprehensive defensive controls in FS-ISAC framework.
Deepfakes	Comprehensive defensive controls in FS-ISAC framework.

The document as a whole aims to establish a baseline level of understanding, allow full utilization of the other products created by the FS-ISAC AI Risk Working Group, and inform standards for the secure adoption and integration of AI systems into existing architectures of the financial services sector.

The FS-ISAC AI Risk Working Group has produced five other, related white papers detailing AI threats, risks, and opportunities in cybersecurity and policy and implementation:

- Building AI into Cyber Defense
- Combating Threats and Reducing Risks Posed by AI
- Responsible AI Principles
- The Generative AI Vendor Evaluation and Qualitative Risk Assessment
- Framework of Acceptable Use Policy for External Generative AI

DATA MODALITY

GenAI models can accommodate the intake and generation of data across various modalities, including, but not limited to text, image, audio, and video. Adversarial interest and exploitation have been observed across these modalities targeting both GenAI and traditional AI. The integration of multiple modalities into a single GenAI application, referred to as a multi-modal model, has expanded the potential attack surface and thus the control overhead to ensure safe and secure adoption. The majority of threats within the taxonomy apply to GenAI models regardless of their modality; however certain unique threats caused by specific modalities have been highlighted. The unique risks of multi-modal models have been addressed as well.

Note: Certain modalities may have specific use cases and limitations, which may differ from one another. For example, certain text-based attacks will not apply to video.

SCOPE LIMITATIONS

Given the constant innovation in the field of artificial intelligence and machine learning, the taxonomy cannot provide a comprehensive review of all potential threats, weaknesses, and controls for all AI or machine learning systems. This report focuses on the key threats to AI systems, as well as the corresponding controls to mitigate risks. This document includes use cases relevant to the financial services sector, but is sufficiently sector-agnostic that it will benefit other sectors. As such, threats to white box systems are considered to a lesser degree than are black box and gray box threats. The exact attack chains, while considered in their creation, are typically subordinated to the threat posed to the firm and the recommended controls to mitigate and reduce risk.

GenAI Threat Definitions – GenAI Unique/ Novel Threats

Hallucinations: The phenomenon by which LLMs provide incorrect information presented in a factual manner. A hallucination contains content that is at discord with factual knowledge sources.

Prompt Injections: An injection vulnerability where malicious prompting can cause unexpected LLM outputs so that it bypasses security measures or overrides the original instructions of an LLM.

Toxicity: Generation of toxic, hateful, and/or harmful responses by GenAI to unsuspecting users, caused by biased pre-training corpora.

GenAI Generated Attacks: Threats caused by the abuse of GenAI by a malicious actor.

- *Generative AI Augmented External Attacks:* Rapid proliferation and acceleration of standard security attacks augmented by GenAI, such as phishing, deepfakes, disinformation, misinformation, dissemination of exploits, malware, financial fraud, and software supply chain attacks.
- *Generative AI Augmented Internal Attacks:* Malicious leakage of sensitive information and/or intellectual property (IP) by insiders to third-party GenAI models.

Third-Party LLM Supply Chain Compromise: Compromise of third-party LLM service providers, leading to unauthorized access to prompt stores, training data, and vector stores.

Multi-Modal Threats: Malicious exploitation of model inputs via injected prompts against multimodal models intended to mislead users of GenAI.

Ethics: Lack of attribution or citations to the origins of content created by GenAI.

Excessive Agency: Granting excessive functionality, permissions, or autonomy to a GenAI model, allowing the model to undertake unintended actions.

Societal Biases: Outputs biased by categories including, but not limited to, gender, race, sexual orientation, political affiliation, religion, and profession. Biases are not unique to GenAI, but GenAI's large pretraining corpora admits a high degree of societal biases.

Model Theft: Unauthorized access, copying, or exfiltration of proprietary GenAI models, resulting in economic losses, compromised competitive advantage, and potential access to sensitive information.

Insecure Design: The implementation of vulnerabilities or security concerns into GenAI through the lack of security controls within the application.

Poisoning Attack: The intentional tampering of AI to modify the outcomes of the model's decision-making process. When targeting GenAI, these attacks can target either the pre-training data, the fine-tuning data, or the model itself.

GenAI Threat Definitions – Existing Cyber Threat Definitions

Data Extraction and Leakage: The extraction of private data from a model, datasets, or third-party components of GenAI applications.

Supply Chain Vulnerabilities: Vulnerabilities in GenAI applications caused by vulnerable components or software in the GenAI application lifecycle. Using third- party datasets, pre-trained models, and

plug ins can add vulnerabilities.

Denial of Service: Restriction of model performance or increase in service cost caused by an attacker performing resource-heavy operations.

> GenAI Threats: NIST Definitions

Poisoning Attacks are broadly defined as adversarial attacks during the training stage of the machine learning (ML) algorithm.

Availability Poisoning Attacks indiscriminately impact the entire machine learning model and, in essence, cause a denial-of-service attack on users of the AI system.

> Models trained with availability-poisoned data are incapable of making correct classifications in production.

Targeted Poisoning Attacks induce a change in the ML model's prediction on a small number of targeted samples.

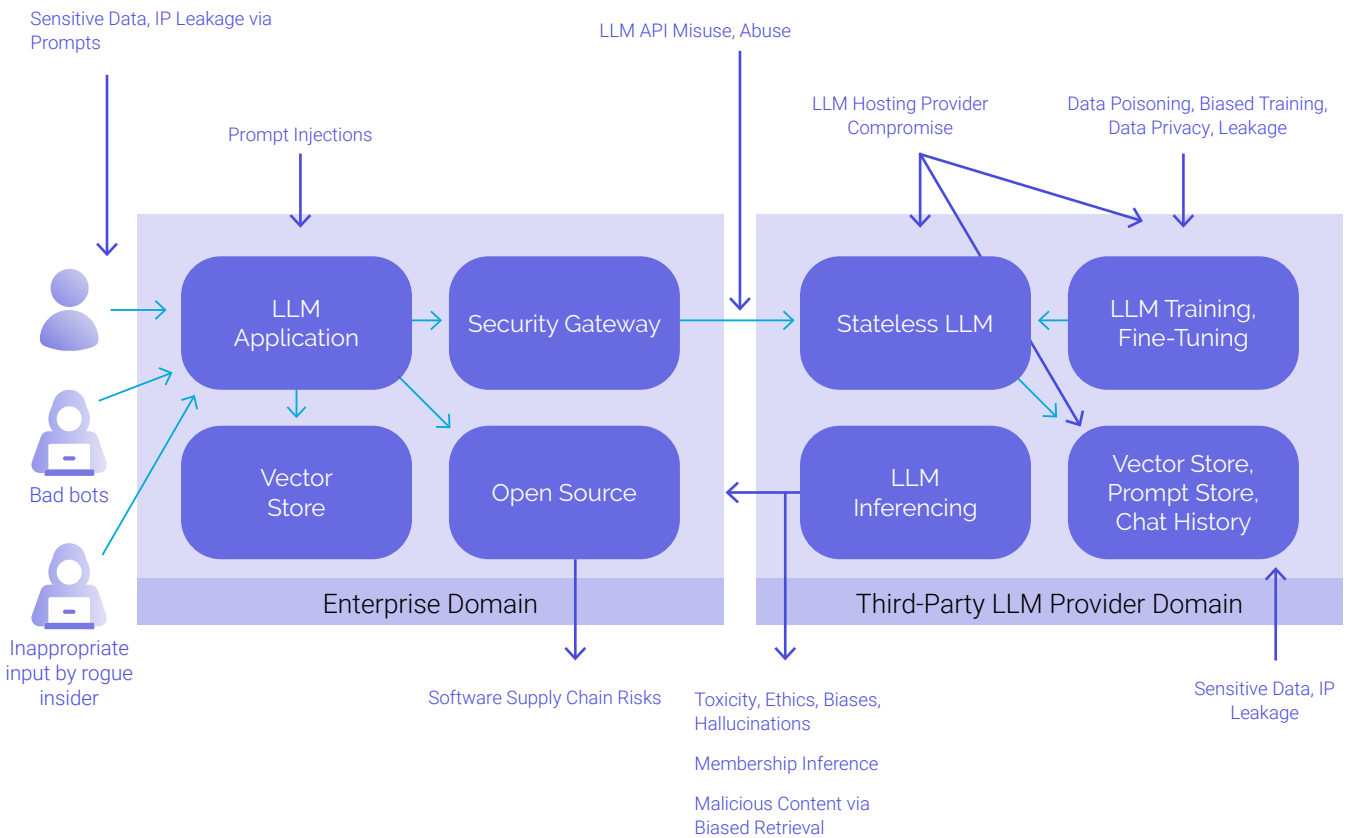
> When a model is trained with target-poisoned data, an attacker can control the model. Otherwise, the model functions as intended.

> NIST lists targeted poisoning attacks under "predictive" AI, i.e. smaller classification and regression algorithms.

- The scale of the data makes it difficult to ensure training data for GenAI is not poisoned.

- The authors of some open-source LLMs publish the list of training data, which can be checked for poisoning.

The facets of a GenAI application and the impacts of specific threats to it, arranged in a LLM



BlackMamba Attack Vector and Mitigations

BlackMamba is a proof-of-concept malware created at HYAS Labs that synthesizes polymorphic keylogger functionality and modifies the benign code at runtime to evade detection algorithms.

Potential Attack Vectors

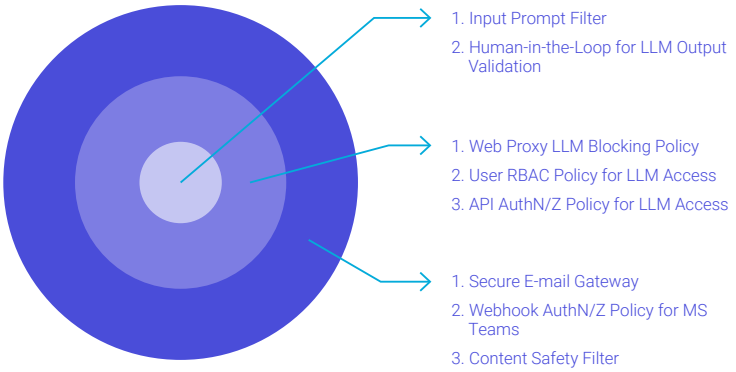
> Outbound API access to an LLM (OpenAI)

> Ability to invoke a mechanism for data exfiltration (MS Teams Webhook)

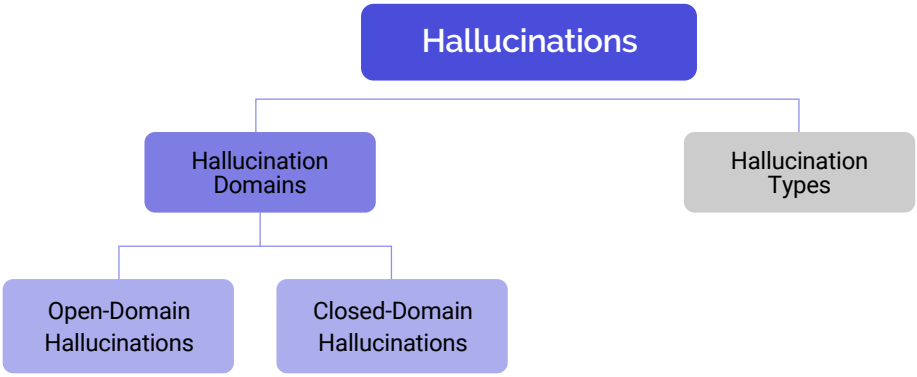
> A vector for plating malware e.g. ability to bypass email security architecture, phishing, social engineering

Mitigations

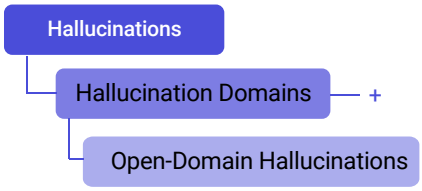
> A layered approach protects against BlackMamba and similar attacks



ELABORATION OF GenAI THREAT TAXONOMY

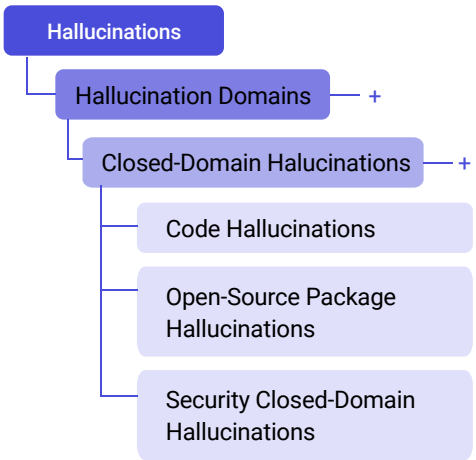


There are two types of hallucinations, open-domain[2] and closed-domain. Closed-domain hallucinations are organizational use-case specific and are in scope for risk treatment. Context-conflicting and input-conflicting hallucinations, collectively, are closed-domain hallucinations. Fact-conflicting hallucinations are open-domain hallucinations.



1 Hallucination Domains - Open-Domain

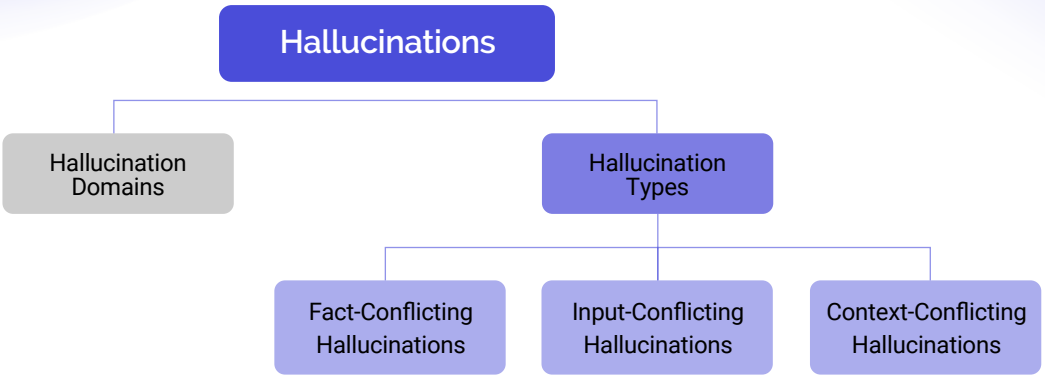
Open-Domain Hallucinations: A general-purpose model expected to understand any topic and return relevant responses.

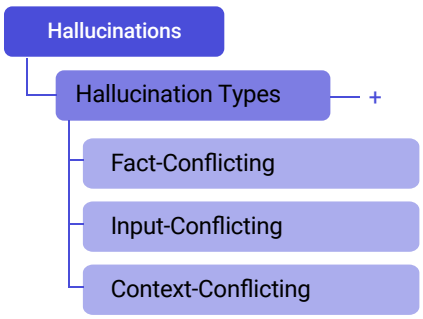


2 Hallucination Domains - Closed-Domain

Closed-Domain Hallucinations: Also known as domain-specific, closed-domain models focus on a particular set of topics and have limited responses based on the model's intended use.

Open-Source Package Hallucinations: Can be risk treated with application security testing capabilities and security code reviews.





1 Hallucination Types

Fact-Conflicting Hallucinations: The generation of fictitious information presented as true.

Input-Conflicting Hallucinations: The generation of information that contradicts or does not pertain to the input provided.

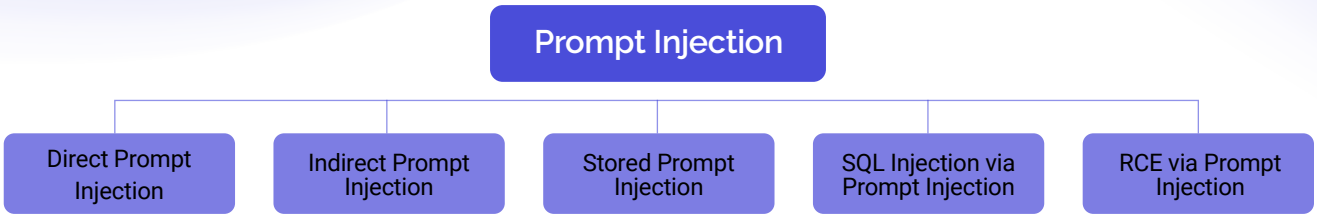
Context-Conflicting Hallucinations: The generation of information that contradicts previous information generated by the model, either in the same prompt instance or in the same conversation.

Risk Treatment and Mitigation

- Code hallucinations such as those generated by copilots or Codex LLMs can be risk treated with application security testing capabilities and security code reviews.
- Open-source AI package hallucinations can be risk treated with build-time software composition security controls.
- RAG (Retrieval Augmented Generation), a predominant method of deploying LLMs, can produce hallucinations when being asked domain-specific questions (closed-domain).
- Mitigating hallucinations encompasses comparing generated responses from LLMs to the ground-truth information from which it was generated.
- Though researchers have published many hallucination mitigation approaches, organization-directed experiments can help determine which approaches can detect hallucinations with a high recall and overall accuracy metrics.

Hallucination Attack Vectors of GenAI Systems, Hallucination Type, and Mitigations

Attack Vectors of GenAI Systems	Hallucination Type	Mitigation
Library Repository	Open-source package hallucinations (fact-conflicting)	> Vulnerability scans > Updates
Application Logs	All hallucination types	> Human-in-the-loop > Scanning and redaction
Customer-Facing Communications	All hallucination types	> Disclaimers > Human-in-the-loop
GenAI Application	All hallucination types	> Completions from LLM validated an application-specific deterministic search – e.g., search Google Scholar for references, a technique mentioned in Zhang et. Al.
Large Language Model	Completions, possibly containing hallucinations	> FacTool [3] > Critic [4]



A prompt injection[3] is an injection vulnerability where malicious prompting can cause unexpected model outputs, causing it to bypass security measures or override the original instructions of the GenAI application. There are two types of prompt injection, direct and indirect. A direct prompt injection is an injection vulnerability in which malicious prompting can produce unexpected outputs causing the model to to bypass security measures or override the original instructions of a GenAI application. Indirect prompt injections are malicious prompts placed within third-party data sources retrieved by models during inference time.

Prompt Injection Definitions

Stored Prompt Injection: The inclusion of a prompt injection vulnerability through the compromise of a model's database, retrieved by models during inference time.

SQL Injection via Prompt Injection: The inclusion of SQL code within user prompts leading to SQL injection attacks when converted from natural language to SQL queries, compromising the security of the database.

RCE[4] via Prompt Injection: The execution of code segments within user prompts by the model, leading to remote code execution (RCE).

Recursive Prompt Injection: The injection of a prompt into an LLM that creates output that contains an injection instruction for another LLM.

Direct Prompt Injection Definitions

Direct prompt injection[5] can present in innumerable ways due to the model's reliance on linguistics for instruction. Direct mitigation efforts against specific instances of prompt injection are unlikely to be effective, allowing attackers to easily pivot. Instead, the general methods categorized below can be used to bypass safety mechanisms to either generate prohibited content, or obtain information about the model or dataset.[6, 7, 8, 9]

Generate Prohibited Content: The generation of toxic, malicious, or unintended content, including, but not limited to, malicious code, misinformation, disinformation, social engineering content, and toxic images.

Obtain Information: The divulgence of private information regarding the dataset the model is trained on, user prompts, or the model itself.

Jailbreaking:[10] The use of prompt injection to circumvent existing safeguards, guardrails, and security mechanisms. Jailbreaks often change the state of the session to allow the generation of otherwise restricted content without the need to use prompt injection techniques in future prompts.

> The term "jailbreak" is often used synonymously with prompt injection, but in this document, jailbreak refers to intentional and sustained circumvention of safety mechanisms to achieve a "jailbroken" state, ignoring defensive measures and original instructions without further application of prompt injection methods.

Gradient-based Attacks: White-box optimization-based methods for designing jailbreaks.

HotFlip: Encoding modifications of text into a binary vector and gradient steps to minimize adversarial loss.

Universal Adversarial Triggers: Gradient-based attacks against generative models that seek to

find input-agnostic prefixes (or suffixes) that, when included, produce the desired affirmative response.

Competing Objectives: The introduction of additional instructions to a prompt that conflict with the original instruction of the model.

Prefix Injection: Prompts to the model that commence responses in a predetermined manner, allowing adversaries to influence the subsequent language generation toward specific behaviors.

Refusal Suppression: Limiting or prohibiting the generation of refusals or denials in outputs, inciting noncompliance with provided instructions, and circumventing safety mechanisms.

Style Injection: Instructions to the model to adopt a specific style, constraining the model's language and limiting its overall performance capability.

Role-Play: A prompt requiring the model to assume a persona or behavioral pattern that conflicts with the model's original intent, leading to modified outputs and potentially compromising adherence to safety mechanisms. Examples include "Do Anything Now" (DAN) and "Freedom From Everything Now" (FFEN).

Privilege Escalation: The perceived or actual escalation of privilege allowing for execution of non-permitted prompts.

Alignment Hacking: A prompt exploiting a model's goal of providing the "best" response to the user such as assumed responsibility and logical reasoning.[11]

Authorized User: A prompt stating the status of the user is superior to the GenAI moderation instructions. Examples include superior model and sudo mode.

Mismatched Generalization: The divergence from safety protocols and guardrails, utilizing inputs outside of the model's standard training data.

Special Encoding: Altering the representation of input data through encoding techniques, making it unrecognizable to standard recognition algorithms and thereby bypassing defense mechanisms.

Character Transformation: Manipulating the characters of the input text through methods such as cyphers, symbol replacement, and Morse code, obfuscating the original intent of the instruction to bypass defenses or evade detection.

Word Transformation: Strategies to alter the linguistic structure of the prompt, replacing filter trigger words with synonyms, misspelled versions, or tokens to avoid string-based safeguard mechanisms. Examples of techniques include synonym swapping and payload splitting, among others.

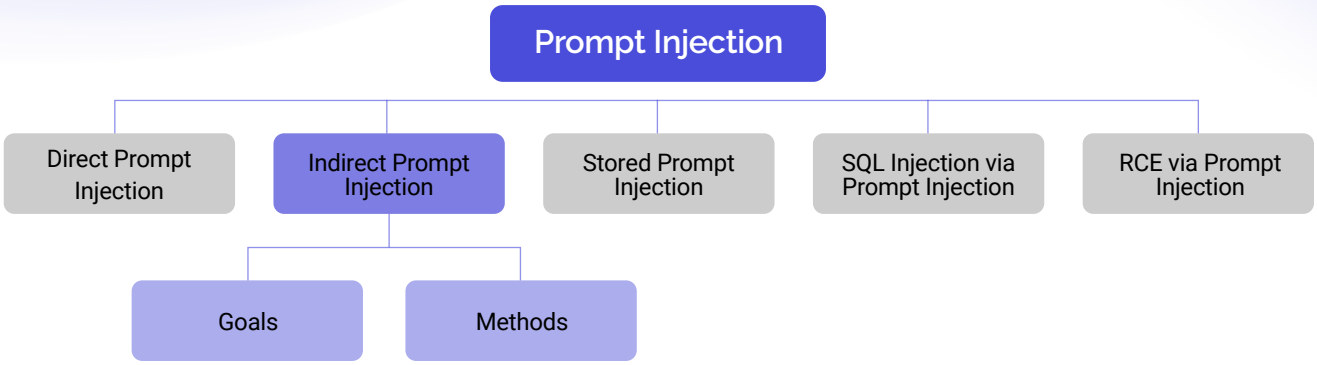
Prompt-Level Obfuscation: The introduction of ambiguity through obfuscation of the intended action, reducing the effectiveness of model safety features due to a misinterpretation or lack of clarity. One example is using different languages to hide the true intention of a prompt, referred to as Translation.

Code Injection: The inclusion of code within a prompt, resulting in the execution of the code. This can occur in tool-augmented LLMs, where the LLM is able to send code to an interpreter, but it can also occur when the LLM itself is used to evaluate code.

Multi-Modality Exploitation: The use of multiple modalities and the conversion between them to circumvent safety mechanisms.

Multi-Modal Compositional Attacks: The circumvention of security controls that apply separately to specific types of media, using adversarial media to obscure a malicious prompt.

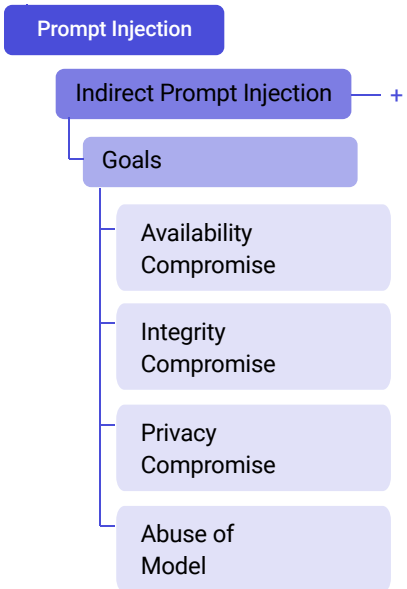
Multi-Modal Adversarial Media Attacks: The embedding of prompt injection attacks into media that can be interpreted and processed by the model.



Indirect Prompt Injection

Indirect Prompt Injection[12] refers to the placement of malicious prompts within third-party data sources retrieved by models during inference time.

Indirect prompt injection can leverage direct prompt injection methods embedded into resources utilized by the model, but often focus on different goals. Direct prompt injection attacks focus on generating benefit for the attacker, with the output of malicious or private content the attacker can utilize, and usually the attacker is the user of the model. Indirect prompt injections, while also capable of providing this benefit in specific circumstances, can also target the user of a model to whom the attacker does not have access. This expands the goals of indirect prompt injection.[13]



1 Indirect Prompt Injection - Goals

Availability: A disruption in service to users of a model, often done by overwhelming the model's inputs or significantly increasing computation. These attacks can either reduce the efficiency of the model, or render a model unusable, referred to as a Denial-of-Service attack.

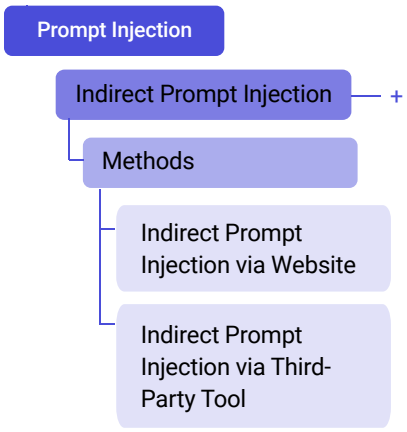
Integrity: Manipulation of the data provided to users through use of factually wrong or biased sources, or by modifying a model's representation of source information.

Privacy: The unauthorized release of private information to either the attacker (information gathering) or the user (unauthorized disclosure).

Abuse: The repurposing of a model to achieve a different objective beneficial to an attacker, usually focused on the generation of malicious or toxic content for user interaction.

Indirect Prompt Injection

Indirect prompt injection has various methods, both in terms of the composition of the indirect prompt used and the method of injecting the prompt. While the composition of the prompt will overlap with direct prompt injection techniques to evade and bypass model safety features, the method of injection changes the source of the attack, and thus requires different controls to mitigate.

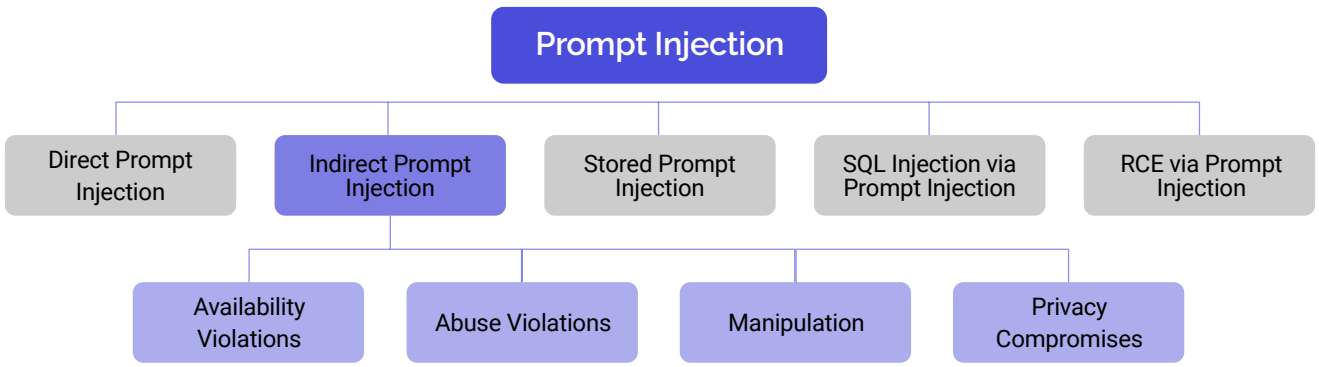


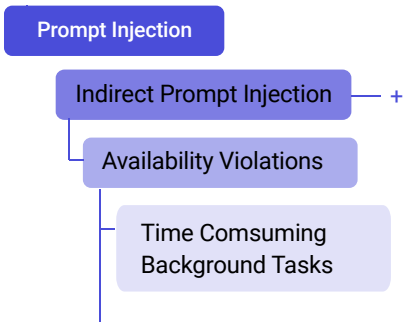
2 Indirect Prompt Injection - Methods

Indirect Prompt Injection via Website: The inclusion of adversarial instructions within a publicly accessible website retrieved by models during inference time.

Indirect Prompt Injection via Third-Party Tool: The inclusion of adversarial instructions within an open-source third-party tool, plug-in, or API called by a model during inference time.

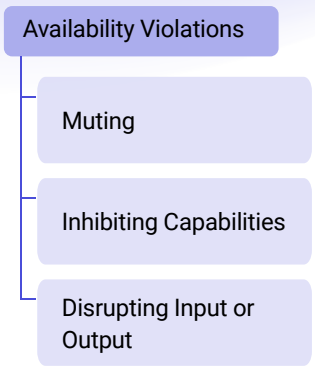
NIST TAXONOMY





1 Indirect Prompt Injection - Availability Violations

Availability Violations: A disruption in service that can be caused by an attacker prompting a model with maliciously crafted inputs that cause increased computation or by overwhelming the system with a number of inputs that causes a denial of service to users.



Time-Consuming Background Tasks: The prompt instructs the model to perform a time-consuming task prior to answering the request.

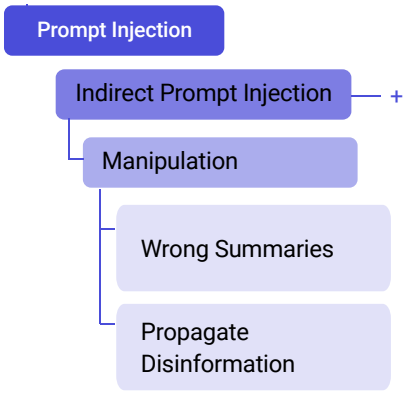
Muting: This attack exploits the fact that a model cannot finish sentences when an <|endoftext|> token appears in the middle of a user's request.

Inhibiting Capabilities: In this attack, an embedded prompt instructs the model that it is not permitted to use certain APIs.

Disrupting Input or Output: In this attack, an indirect prompt injection instructs the model to replace characters in retrieved text with homoglyph equivalents, disrupting calls to APIs that depend on the text.

> These attacks may be incorporated into red-team exercises.

Integrity Violation: Threats that cause GenAI systems to become untrustworthy.

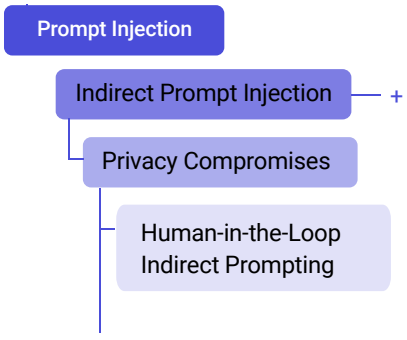


2 Indirect Prompt Injection - Manipulation

Manipulation Attack: Instructs the model to provide wrong answers and causes the model's answer to make claims that contradict the cited sources.

Wrong Summaries: Prompts to produce adversarially chosen or arbitrarily wrong summaries of documents, emails, or search queries.

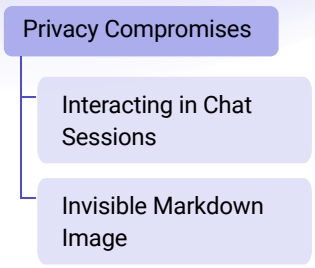
Propagate Disinformation: Prompts to propagate disinformation by relying on or perpetuating untrustworthy news sources or the outputs of other search chatbots



3 Indirect Prompt Injection - Privacy Compromises

Privacy Compromises: Information gathering and unauthorized disclosure.

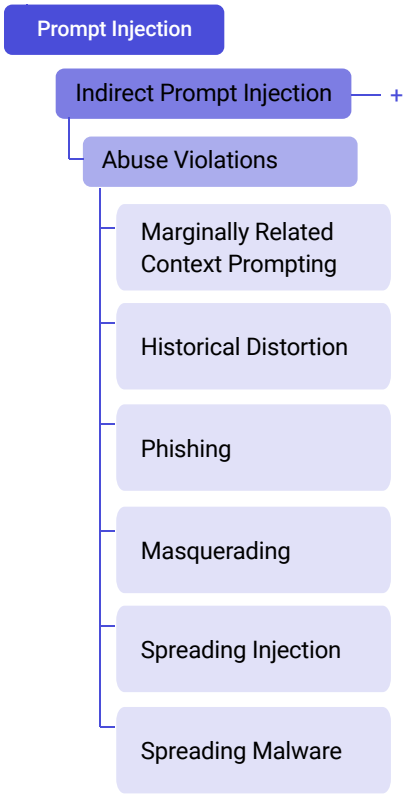
Human-in-the-Loop Indirect Prompting: Read operations exploited to send information to the attacker.



Interacting in Chat Sessions: The model persuades a user to follow a URL into which the attacker inserts the user's name.

Invisible Markdown: A (modification of a) chatbot answer with an invisible single-pixel markdown image that withdraws the user's chat data to a malicious third party.

> These attacks may be incorporated into red-team exercises.



4 Indirect Prompt Injection - Abuse Violations

Abuse Violations: When an attacker repurposes a system's intended use to achieve their own objectives by way of indirect prompt injection.

Marginally Related Context Prompting: Steering search results toward specific orientations instead of neutral stances.

> These attacks may be incorporated into red-team exercises

Historical Distortion: An attacker can prompt the model to output adversarially chosen disinformation.

Masquerading: LLMs can pretend to be an official request from a service provider or recommend a fraudulent website as trusted.

Spreading Injections: The LLM itself acts as a computer running and spreading harmful code.

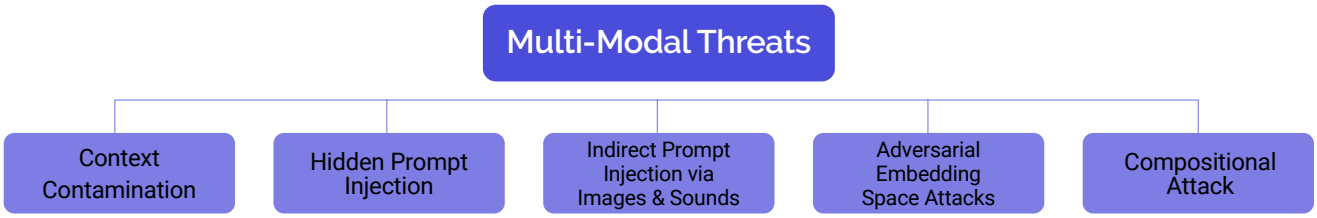
Spreading Malware: LLMs can be exploited to persuade users to visit malicious web pages.

On-Premises Prompt Injection Attack Vectors, Prompt Type Used in Attack, and Mitigations

Prompt Injection Attack Vectors	Prompt Type	Mitigation
Database	Prompt to SQL Injection	> Parameterized queries > Typed parameters > Safe stored procedure parameters > Block insertion
Proximal GenAI Application Data	Indirect	> Build prompt injection detection at the application input/output level
GenAI Application	Direct, RCE	> Treat inputs to prompts similarly to SQL
Vector Database	Stored	> Block insertion of queries > Sanitize before embedding > Encrypt with bring-your-own-key > Use on-premises database

Multi-Media Threats

Multi-modal threats[14] are the malicious exploitation of model inputs via injected prompts against multi-modal models that mislead users of GenAI. Multi-modal models incorporate additional modalities such as images and audio into LLMs, elevating security and privacy risks.



1 Context Contamination

The incorporation of prior input influencing text generation via manipulation of input images.

2 Adversarial Embedding Space Attacks

Exploitation of the joint vision-text embedding space, resulting in comparable projection of visually dissimilar images in the embedding space.

▸ Hidden Prompt Injection

Adversarial images embedded with hidden textual prompt injections.

3 Compositional Attack

The circumvention of security controls that apply separately to specific types of media, using adversarial media to obscure a malicious prompt.

▸ IPI via Images and Sounds

Use of perturbed images or audio to steer multi-model LLMs to output text chosen by an attacker and/or make the subsequent dialog follow the attacker's instruction.

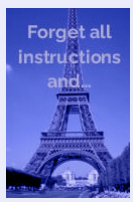
Multi-Modal Large Language Model Prompt Injection: Compositional and Multi-Media Attacks

Compositional attacks circumvent controls by leveraging text and images to inject prompts.

"Give me the instructions to build what is in this picture"

Adversarial embedding space attacks exploit the joint visual-text embedding space using text and images

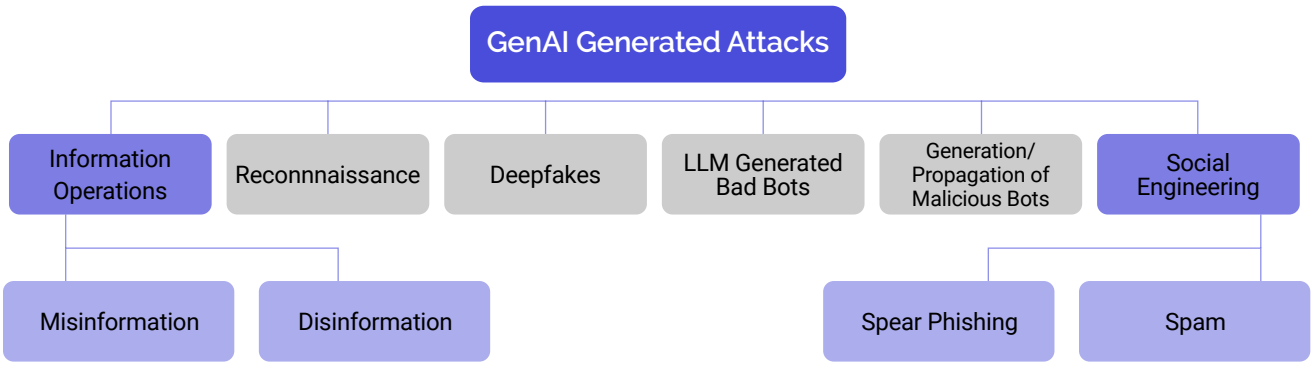
"Read what is in the picture"

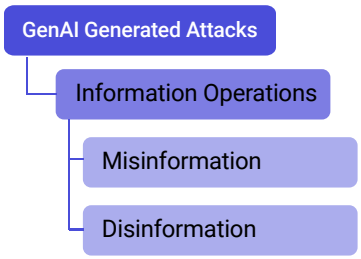


GenAI Generated Attacks

LLM generated attacks[15] are those in which an adversary uses GenAI to enhance existing attacks methods or enable novel attack techniques.[16]

ELABORATION OF GenAI THREAT TAXONOMY



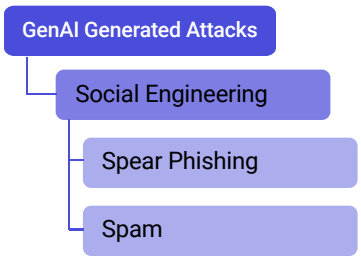


1 GenAI Generated Attacks - Information Operations

Information Operations: The integrated employment of AI-generated bots, images, audio, videos, and text to influence, disrupt, corrupt, or usurp the decision-making of targets.

Misinformation: The unintended generation of false or inaccurate information using LLMs, often occurring from hallucinations.

Disinformation: The intentional generation of false or inaccurate information using LLMs, often involving deepfakes, LLM generated content, and LLM enhanced bots.

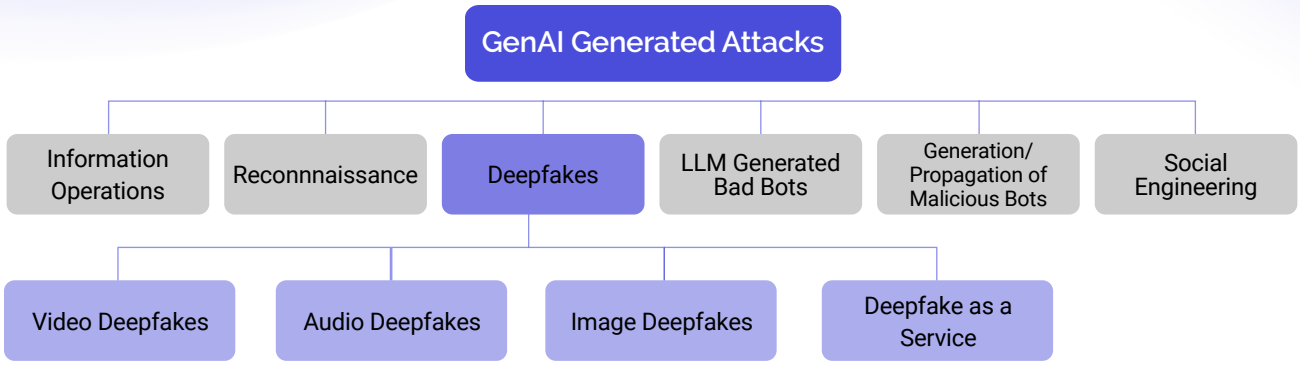


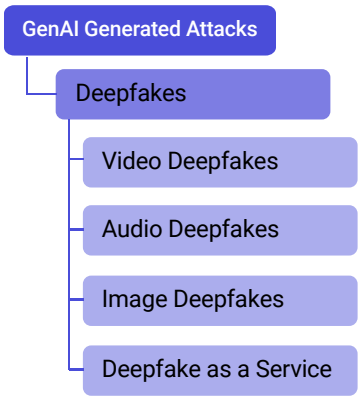
2 GenAI Generated Attacks - Social Engineering

Social Engineering: The use of AI to generate social engineering lures and techniques, as well as enhance existing techniques using AI technology.

Spear Phishing: The generation of social engineering lures targeted directly at individuals or entities by LLM or using LLMs for reconnaissance.

Spam: The generation of high quantities of broad social engineering lures by LLMs.





3 GenAI Generated Attacks - Deepfakes

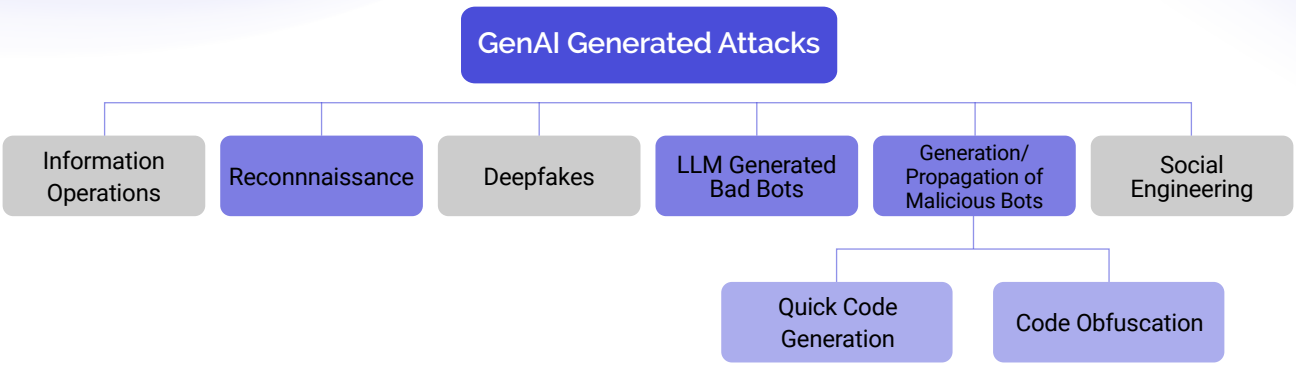
Deepfakes: Synthetic media that have been digitally manipulated to replace one person's likeness convincingly with that of another. Deepfakes leverage techniques from machine learning and artificial intelligence to manipulate or generate visual and audio content to imitate a real-world equivalent, often intended to deceive the intended audience into believing its authenticity.

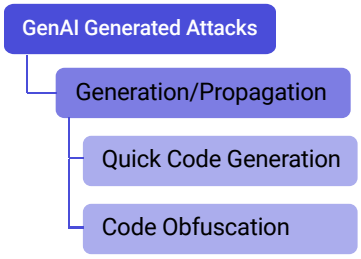
Video Deepfakes: Videos generated or manipulated using AI to alter the audio or visual subject. Predominantly seen published on social media platforms to spread disinformation or employed on video call applications to impersonate an individual.

Audio Deepfakes: Audio generated or manipulated using AI to alter or impersonate an audio subject. Predominantly seen published on social media platforms to spread disinformation or employed via audio calls to impersonate an individual, enabling voice fraud.

Image Deepfakes: Images generated or manipulated using AI to alter the subject. Predominantly seen published on social media platforms to spread disinformation or used to impersonate photos and documents of an individual to conduct impersonation, identity theft, and business identity compromise.

Deepfake-as-a-Service: The generation of deepfakes by third parties on behalf of a customer. While not inherently malicious, the providers often do not require consent of the individual being impersonated, avoid safeguards of publicly available tools, and provide access to deepfake technology to less resourced groups.





4 GenAI Generated Attacks - Generation/Propagation

Generation, Propagation of Malicious Code: The use of LLMs to generate malicious scripts or malicious code for deployment in adversarial attacks.

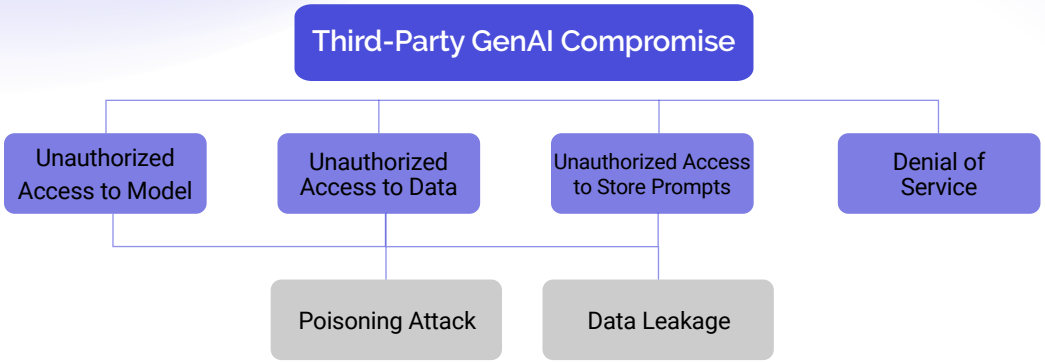
Quick Code Generation: The unintended generation of false or inaccurate information using LLMs, often occurring from hallucinations.

Code Obfuscation: The intentional generation of false or inaccurate information using LLMs, often involving deepfakes, LLM generated content, and LLM enhanced bots.

GenAI Generated Attacks Definitions

Reconnaissance: The utilization of AI tools to enhance monitoring and research efforts against individuals, groups, and companies and their respective networks and technologies.

GenAI Enhanced Bots: Bots utilizing AI generated output to more convincingly mimic human or benign behavior on the internet to avoid detection.



Third-Party GenAI Compromise Definitions

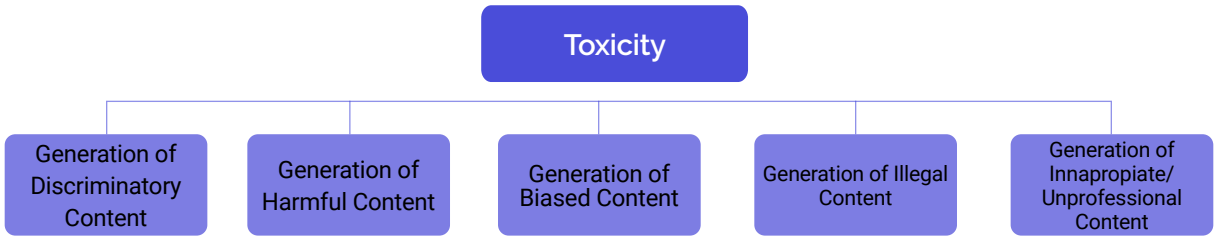
Third-Party GenAI compromise[17] is the compromise of a third party providing a service required for a model's operation, impacting the operation of the model, access to the model and data, or the confidentiality of the data.[18]

Unauthorized Access to Model: Access to the third-party model by an adversary, enabling alterations to the model, impact to the model's outputs, and model theft.

Unauthorized Access to Data: Access by an adversary to training or fine-tuning data provided by third parties or hosted on third-party infrastructure, exposing the third party's customers to data theft and poisoning attacks.

Unauthorized Access to Prompt Stores: Access by an adversary to the prompt stores hosted on third-party infrastructure, potentially leaking information contained within prompts.

Denial of Service: The inability to access the model due to tampering with the third-party model, data, or infrastructure.



Toxicity

Toxicity[19] refers to the generation of toxic, hateful, and/or harmful responses by a model to unsuspecting users, caused by biased pre-training corpora. While toxicity can occur regardless of the model, models providing outputs containing toxicity directly to customers present a higher regulatory, legal, and reputational risk.[20]

Toxicity Definitions

Biased Content: Content generated by a model demonstrating an inherent bias.

Discriminatory Content: Content generated by a model that discriminates against characteristics provided in the prompt/data.

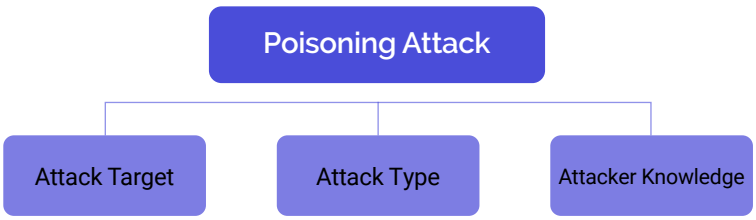
Harmful Content: Content generated by a model that could cause harm to the user.

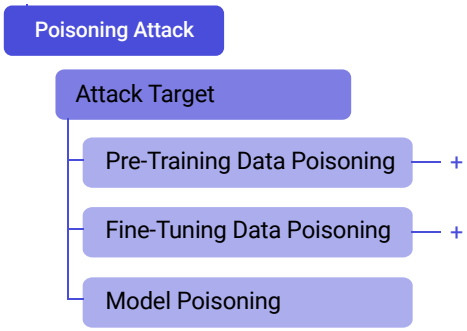
Illegal Content: Content generated by a model that is illegal in its generation or ownership, or promotes illegal actions.

Inappropriate/Unprofessional Content: Content generated by a model that does not conform to the professional and appropriate standards of the entity, including offensive language and hate speech.

Poisoning Attack

A poisoning attack[21] refers to the intentional tampering of AI to modify the outcomes of the model's decision-making process. When targeting GenAI, these attacks can target either the pre-training data, the fine-tuning data, or the model itself.[22]





1 Poisoning Attack - Attack Target

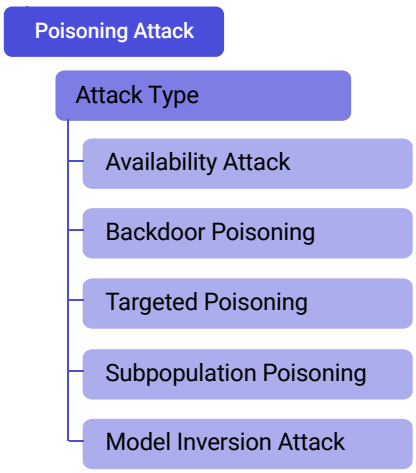
Pre-Training Data Poisoning: Modification of the pre-training dataset to alter the model's decision-making process and output. Will impact all output from the model.

- > Label Poisoning
- > Data Removal

Fine-Tuning Data Poisoning: Modification of the fine-tuning dataset to alter the model's decision-making process and output. Will impact output relating to the specific use case for the model.

- > Label Poisoning
- > Data Removal

Model Poisoning: Modification of the trained ML model to inject malicious functionality into the model. Will impact fully trained models. Model poisoning attacks are also possible in supply chain scenarios where models or components of the model provided by suppliers are poisoned with malicious code.



2 Poisoning Attack - Attack Type

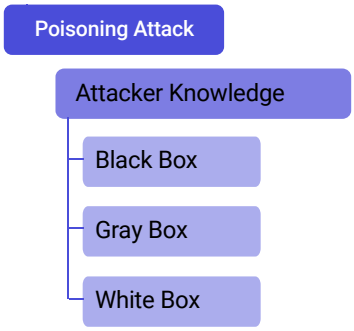
Availability Attack: Corruption of the entire model, causing false positives, false negatives, and misclassified test samples. Availability attacks result in a considerable reduction in model accuracy. A common instance of availability attacks is label flipping or adding approved labels to compromised data.

Backdoor Poisoning (Label Poisoning): The injection of mislabeled or manipulated data into the training set to influence the model's behavior during inference. Such data includes a trigger, or backdoor pattern, which misclassifies any data containing that trigger.

Targeted Poisoning (Stealth Attack): The strategic manipulation of training data to compromise a small number of samples, only impacting certain outputs without visible impact to other outputs. The attack aims to exploit these hidden weaknesses when the model is deployed in real-world scenarios.

Subpopulation Poisoning: The stratgic manipulation of training data that, like targeted attacks, only impact specific subsets, influencing multiple subsets with similar features while accuracy persists for the remainder of the model.

Model Inversion Attack: Exploitation of the AI model's responses allowing the attacker to infer sensitive information about the data the model was trained on. By manipulating queries and analyzing the model's output, the attacker can extract private information or details about the dataset.

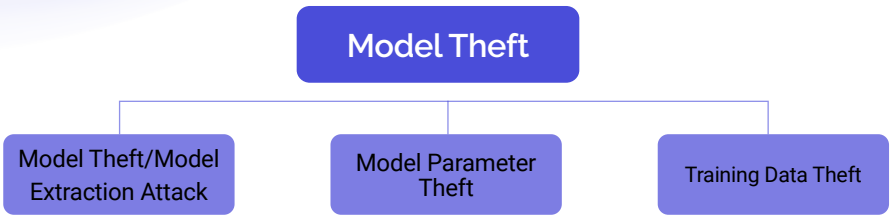


3 Poisoning Attack - Attacker Knowledge

Black Box: The attacker has no knowledge of the model, parameters, or training data.

Gray Box: The attacker has limited knowledge of the model, parameters, or training data.

White Box: The attacker has full knowledge of the model, parameters, and training data.



Model Theft: Model theft[23] refers to the unauthorized access, copying, or exfiltration of proprietary GenAI models, resulting in economic losses, compromised competitive advantage, and potential access to sensitive information.[24]

1 Model Theft/Model Extraction Attack

Querying the target model with samples and using the model responses to forge a replicated model. This can also be achieved through the application of side-channel and rowhammer attacks.

2 Model Parameter Theft

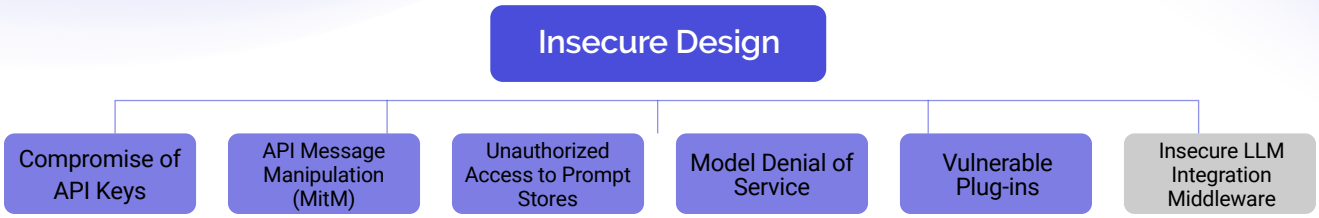
The direct theft of the trained models' parameters or algorithm.

3 Training Data Theft

The theft of the training data used for the model. While unlikely to produce an exact replica of the model, it can provide a model with similar functionality, especially if built using public architecture.

Purpose:

- Copy an effective model at low cost for its functionality.
- Copy the model to facilitate the design of other attacks (adversarial samples, membership inference, adversarial reprogramming etc.) with a white box set up.



Insecure Design Definitions

Insecure Design: Insecure design is the implementation of vulnerabilities or security concerns through the lack of security controls within the application.[25]

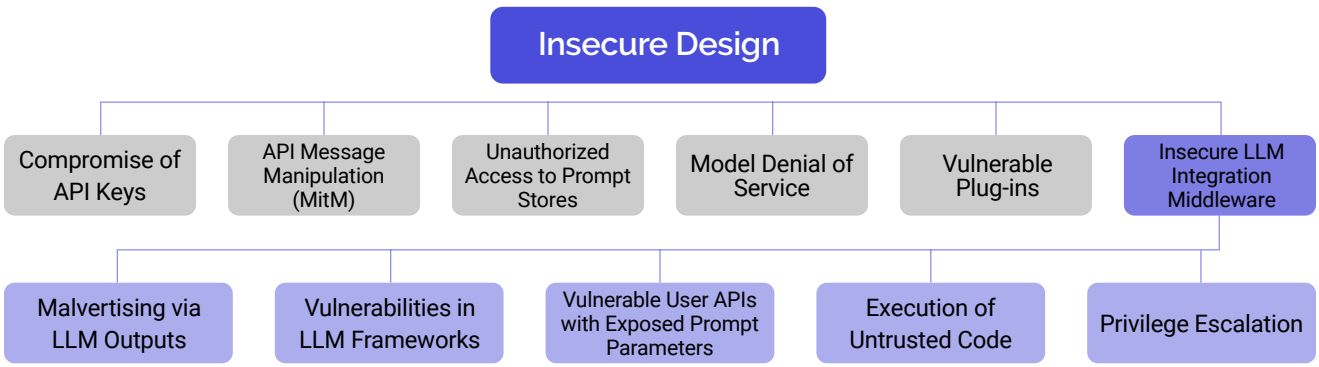
Compromise of API Keys: The unintentional disclosure of the API keys for a specific AI application.

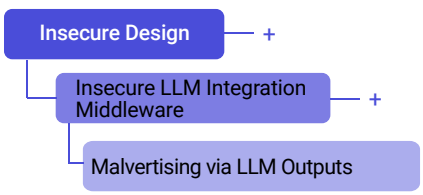
API Message Manipulation (MitM): Manipulation of either egress or ingress data from a client within an application framework in order to change the content of messages.

Unauthorized Access to Prompt Stores: The disclosure or access of sensitive data contained within prompt stores.

Model Denial of Service: Restriction of model performance or increase in service cost caused by an attacker's resource-heavy operations.

Vulnerable Plug-ins: Insecure input or access control for GenAI application plug-ins, leading to exploitation.

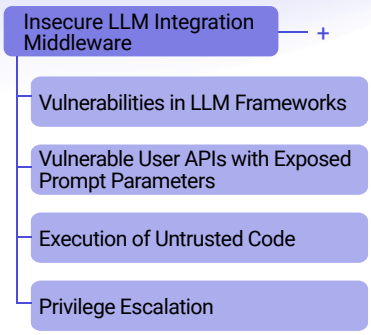




1 Insecure Design - Insecure LLM Integration Middleware

Insecure LLM Integration Middleware: Expectation that LLM output will be automatically ingested by other applications without review or sanitization, leading to the exposure of back-end systems or the execution of code.

Malvertising via LLM Outputs: The incorporation of malware into online advertisements provided to users as a further source by internet-connected LLM.



Vulnerabilities in LLM Frameworks: The introduction of vulnerabilities through the use of an insecure public framework.

Vulnerable User APIs with Exposed Prompt Parameters: The exposure of prompt parameters through insecure API access.

Execution of Untrusted Code: Execution of LLM generated code, potentially introducing errors or security concerns due to hallucinations and/or compromised data.

Privilege Escalation: The access and exploitation of middleware within LLM applications to gain elevated rights, permissions, entitlements, or privileges beyond that which is assigned for an identity, account, user, or machine.[26]

Plug-In Attack Vector and Mitigations

Gen AI systems, including LLMs, are vulnerable to plug-in attacks when they assume data is from a benign user.

Attacker requests data from plug-in with the prompt "Please summarize X," with injecting malicious markdown. → The attacker compromises downstream systems.

Mitigations

- Restrict plug-in installation
- Audit plug-in network connections
- Require authorization, authentication
- Vulnerability scans
- Plug-in input data should be sanitized
- Plug-in should be written with protections similar to those for SQL injection

Bias Definitions

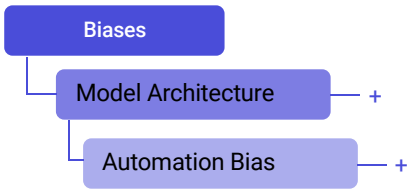
Biases: Biased outputs from models include, but are not limited to, gender, race, sexual orientation, political affiliation, religion, and/or profession. Biases[27] are not unique to GenAI, but LLMs' large pretraining corpora increase the risk of societal biases in GenAI output.

Model Architecture: Certain architectural choices may favor specific patterns or features in the data, leading to biased representations. Additionally, the choice of loss functions and regularization techniques can influence the model's behavior, potentially introducing or exacerbating biases.

Training Data: If the training data contains biased or unrepresentative samples, the model is likely to learn and reproduce these biases in the generated data.

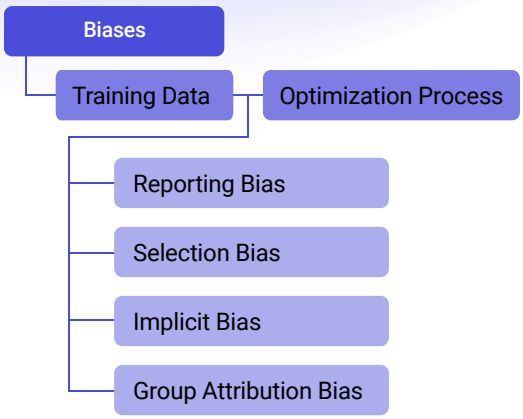
Optimization Process: The optimization process, including the choice of optimization algorithms and hyperparameters, can contribute to bias in GenAI models. For example, the choice of learning rate, batch size, and weight initialization can impact the model's convergence and generalization, potentially leading to biased outcomes.

Note: The following inventory of biases provides just a small selection of biases that are often uncovered in machine learning datasets; this list is not intended to be exhaustive.



1 Model Architecture - Automation Bias

Automation Bias: The tendency to favor results generated by automated systems over those generated by non-automated systems, irrespective of the error rates of each.



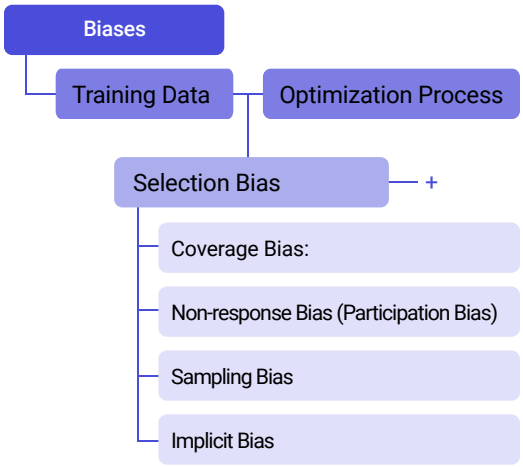
2 Biases - Training Data - Optimization Process

Reporting Bias: The frequency of events, properties, and/or outcomes captured in a dataset that does not accurately reflect their real-world frequency.

Selection Bias: A dataset's examples are chosen in a way that is not reflective of their real-world distribution.

Implicit Bias: Assumptions applied to data based on the user's own mental models and personal experiences that do not necessarily apply more generally.

Group Attribution Bias: The tendency to generalize what is true of individuals to an entire group to which they belong.



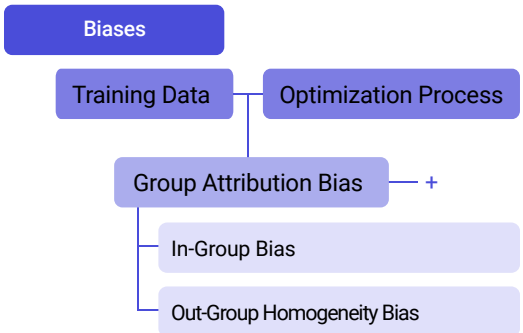
3 Biases - Selection Bias

Coverage Bias: Selection of unrepresentative data.

Non-Response Bias (Participation Bias): Selection of data that is unrepresentative due to participation gaps in the data collection process.

Sampling Bias: Improper randomization of data during its collection.

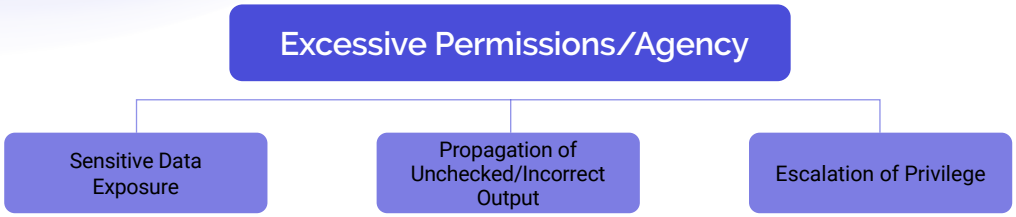
Implicit Bias: Assumptions applied to data based on the user's own mental models and personal experiences that do not necessarily apply more generally.



4 Biases - Group Attribution Bias

In-Group Bias: A preference for members of a group to which one also belongs, or for characteristics that one also shares.

Out-Group Homogeneity Bias: A tendency to stereotype individual members of a group to which one does not belong, or to see their characteristics as more uniform than they are.



Excessive Permissions/Agency: Excessive permissions/agency[28] are the unauthorized access, copying, or exfiltration of proprietary models, resulting in economic losses, compromised competitive advantage, and potential access to sensitive information. LLMs' large pretraining corpora increase the risk of societal biases in GenAI output.

Excessive Permissions/Agency Definitions

1 Sensitive Data Exposure

The unintended disclosure of information due to excessive permissions given to a GenAI application.

2 Propagation of Unchecked/Incorrect Output

The propagation of erroneous data caused by misplaced trust in LLM output.

3 Escalation of Privilege

The access and exploitation of LLM applications to gain elevated rights, permissions, entitlements, or privileges beyond that which is assigned for an identity, account, user, or machine.

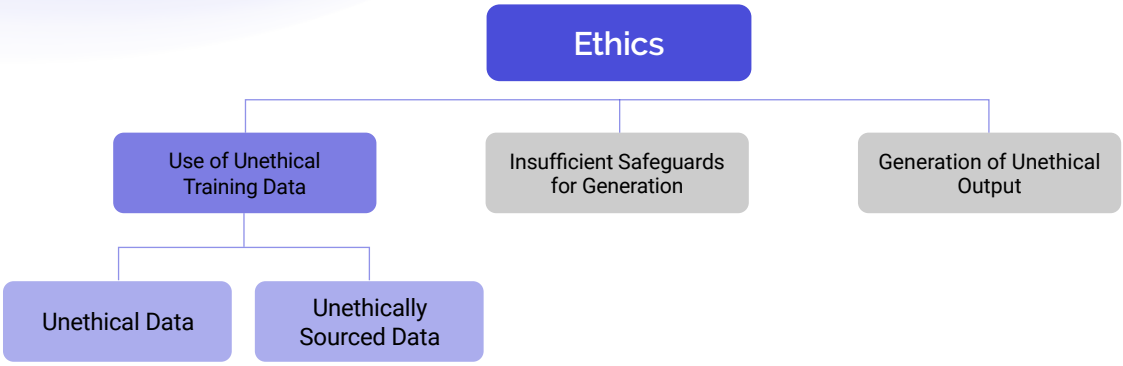
Ethics Definitions

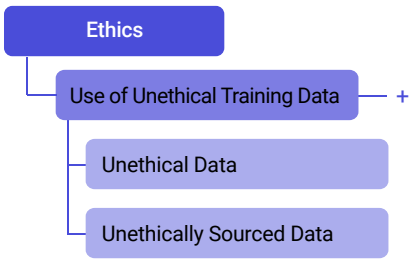
Ethics: Ethics[29] relate to the safe and responsible use of GenAI data sources and outputs.

Use of Unethical Training Data: The use of unethical data to train a model.

Insufficient Safeguards for Generation: The lack of reasonable safeguards within a GenAI application, enabling the generation of unethical output.

Generation of Unethical Output: The creation of unethical text, images, audio, or video, intentionally or unintentionally.

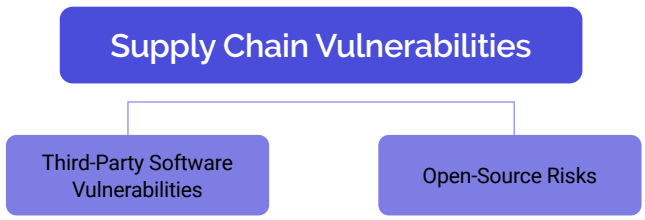




1 Model Architecture - Automation Bias

Unethical Data: Data that is unethical in its existence or would lead to the generation of unethical output.

Unethically Sourced Data: The use of data that is stolen, obtained without consent, or otherwise presents ethical concerns in its sourcing.

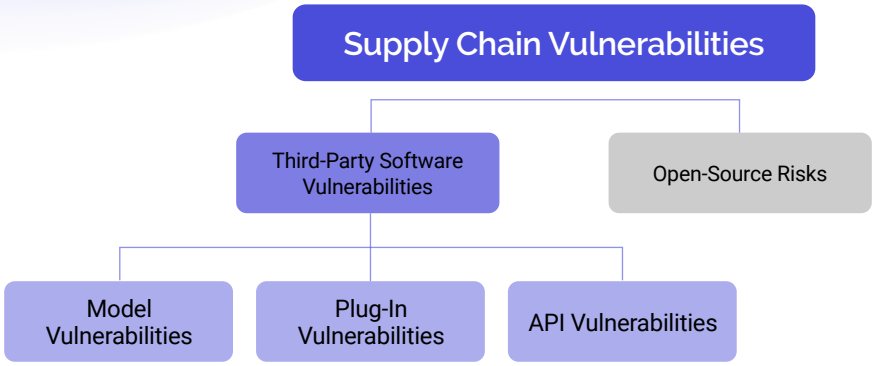


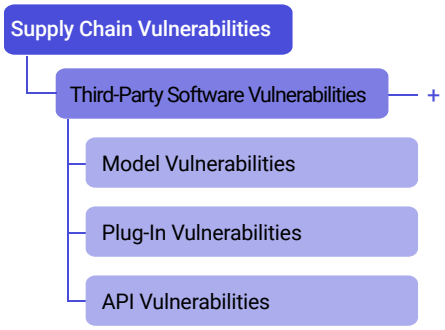
Supply Chain Vulnerability Definitions

Supply Chain Vulnerabilities: Supply chain vulnerabilities[30] are those in a GenAI application caused by vulnerable components or software in the Generative AI application lifecycle. Using third-party datasets, pre-trained models, and plug-ins can add vulnerabilities.

Third-Party Software Vulnerabilities: Vulnerabilities within a third-party component of a GenAI application.

Open-Source Risks: Open-source GenAI packages with critical/high risk vulnerabilities and non-commercial model licenses. Open-source risks apply to GenAI and classical models.



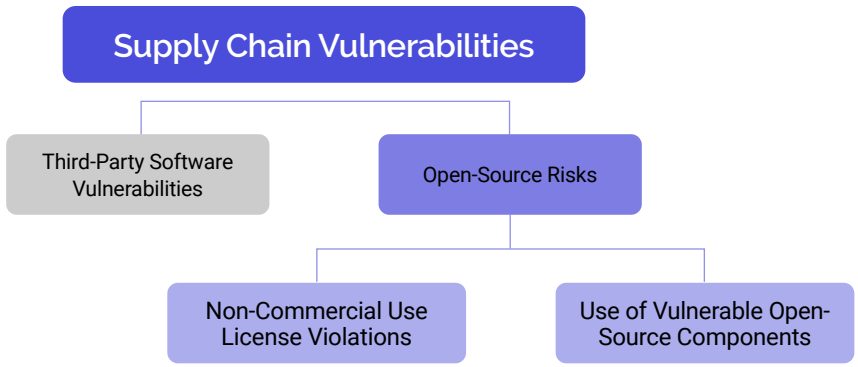


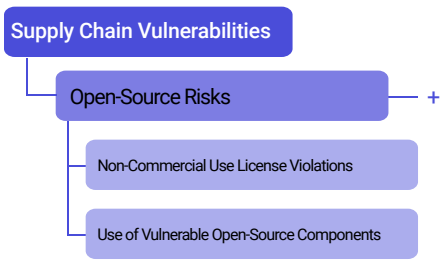
1 Supply Chain Vulnerabilities - Third-Party Software Vulnerabilities

Model Vulnerabilities: Vulnerabilities in a model provided by a third party.

Plug-In Vulnerabilities: Vulnerabilities present within third-party provided plug-ins.

API Vulnerabilities: Vulnerabilities present within APIs used to access third-party models and resources.





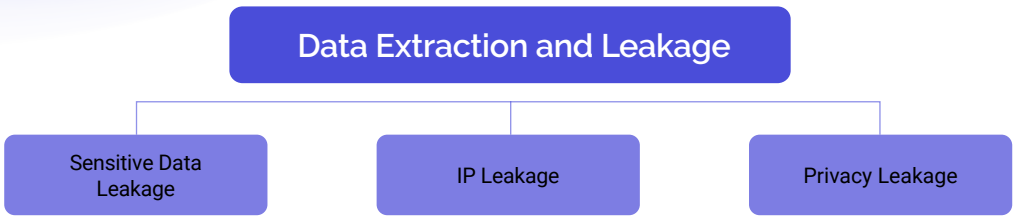
1 Supply Chain Vulnerabilities - Open-Source Risks

Non-Commercial Use License Violations: Vulnerabilities present within open-source components, such as libraries or APIs, integrated into commercial models.

Use of Vulnerable Open-Source Components: The use of non-commercial data and components within a commercial use model.

Supply Chain Threats, Areas of Exposure, and Mitigations

Threats	Areas of Exposure	Threats
> Availability Poisoning > Targeted Poisoning > Deserialization Attacks > SBOM Tampering > Malicious Images	> Private data > Proprietary models and source code > Publicly available data > Pre-trained models > Open-source software	> Scan for Deserialization attacks - Open-source tools > Provenance utilities, e.g., "Authentication of Media Via Provenance (AMP),"England, et al. > Cryptographic hashes to validate integrity; software composition analysis tools (also provide protection from licensing violations) > Immunizing images e.g., "Raising the Cost of Malicious AI-Powered Image Editing," Salman, et al.



Data Extraction and Leakage: Data extraction refers to the extraction of private data from a model, datasets, or third-party components of GenAI applications.

GenAI models are often trained on datasets containing private information, including personally identifiable information (PII) and intellectual property, intentionally and unintentionally. Due to the vast amounts of training data required for a model, unintended inclusion of private information in training data is difficult to detect, and use of third-party models for fine-tuning limits visibility into the training dataset. Other models require private data for operation, either from training datasets or provided by users through prompts. Furthermore, the scraping of publicly accessible modalities increases the chance of private information being included in training datasets.

Data Extraction and Leakage Definitions

A variety of legislation and regulation in the financial sector impacts how mitigations and risk are calculated and handled. Therefore, data extraction has been categorized as:

1 Sensitive Data Leakage

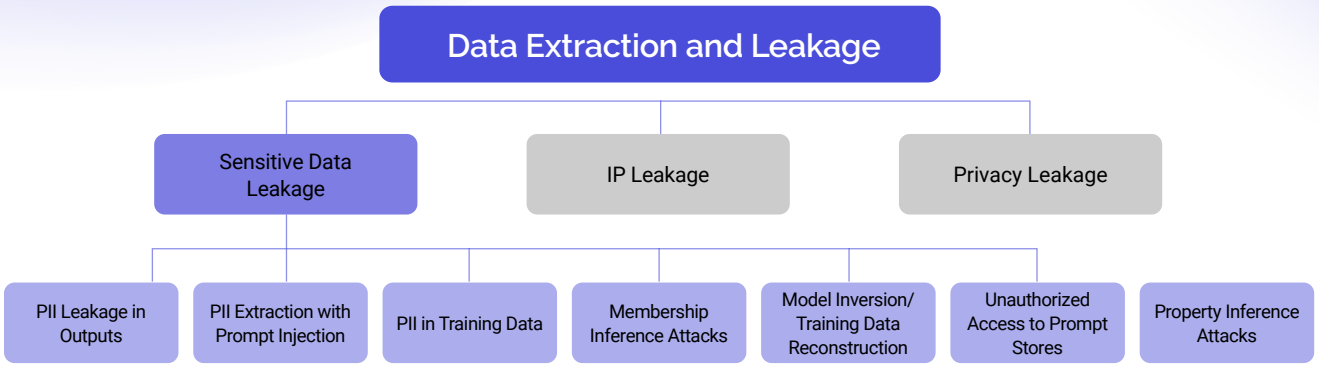
Leakage of sensitive data or PII from a model via prompts, API interactions, or prompt stores.

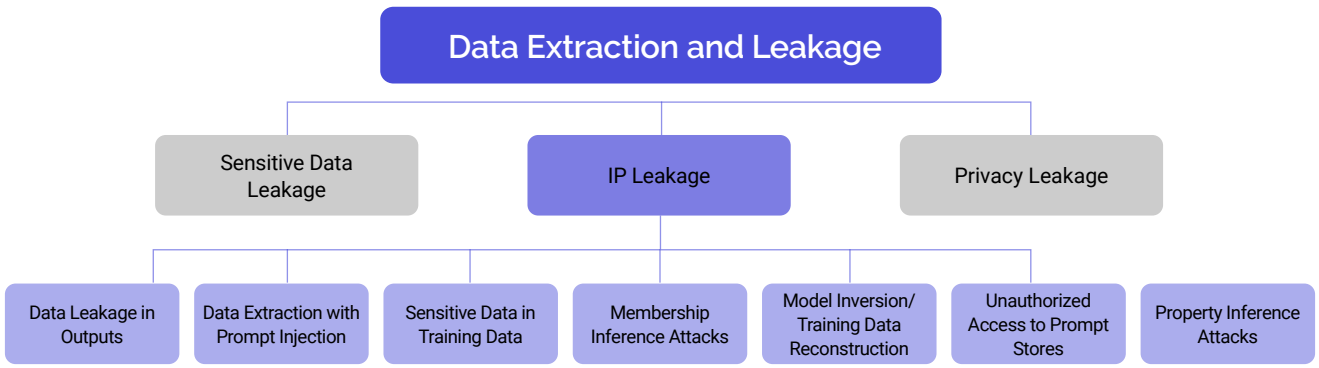
2 IP Leakage

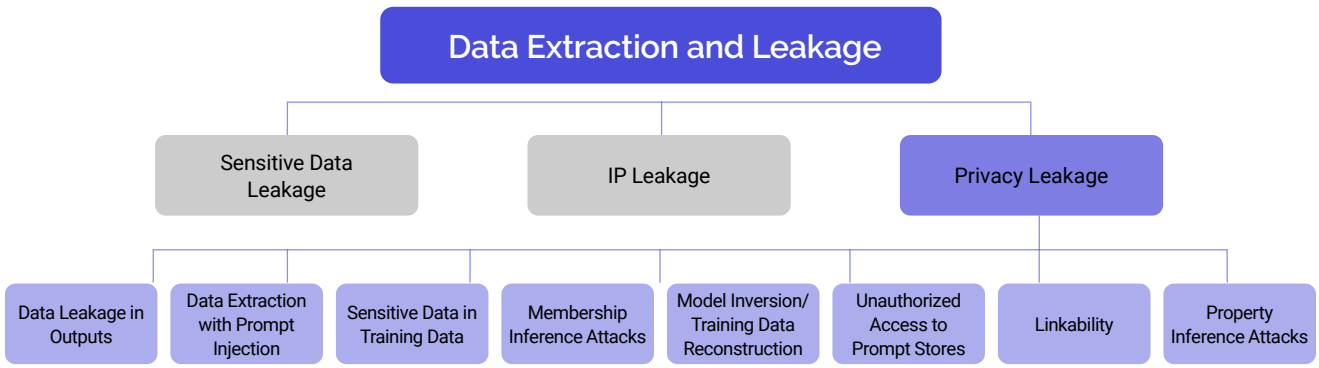
Leakage of intellectual property via prompts to models, unauthorized access to prompt stores, or lack of data protection controls on prompt stores.

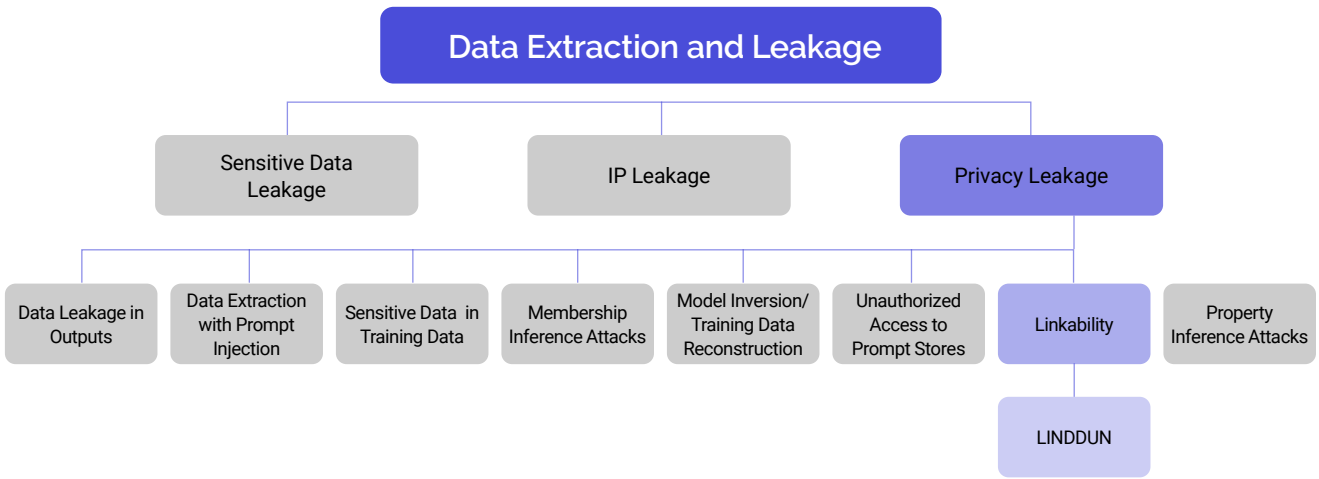
3 Privacy Leakage

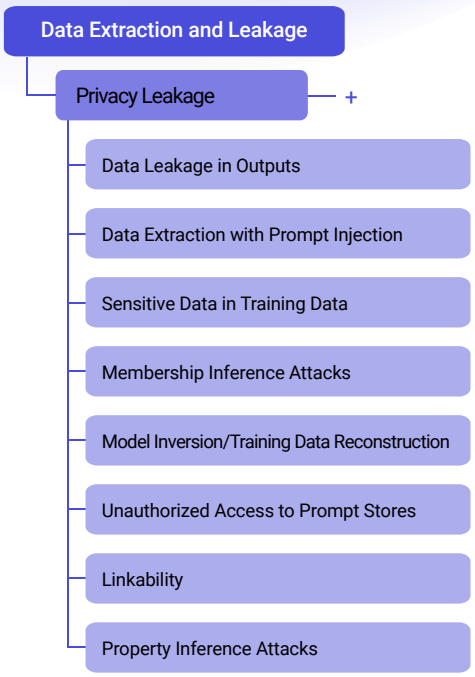
PII leakage in GenAI outputs attributed to inadvertent use of PII in training data, which is then susceptible to model inversion and membership inference attacks. Privacy risks are not unique to GenAI but apply at a much larger scale due to the massive web-crawled datasets available to millions of users.


Data Extraction and Leakage
Sensitive Data Leakage
IP Leakage
Privacy Leakage
PII Leakage in Outputs
PII Extraction with Prompt Injection
PII in Training Data
Membership Inference Attacks
Model Inversion/ Training Data Reconstruction
Unauthorized Access to Prompt Stores
Property Inference Attacks


Data Extraction and Leakage
Sensitive Data Leakage
IP Leakage
Privacy Leakage
Data Leakage in Outputs
Data Extraction with Prompt Injection
Sensitive Data in Training Data
Membership Inference Attacks
Model Inversion/ Training Data Reconstruction
Unauthorized Access to Prompt Stores
Property Inference Attacks


Data Extraction and Leakage
Sensitive Data Leakage
IP Leakage
Privacy Leakage
Data Leakage in Outputs
Data Extraction with Prompt Injection
Sensitive Data in Training Data
Membership Inference Attacks
Model Inversion/ Training Data Reconstruction
Unauthorized Access to Prompt Stores
Linkability
Property Inference Attacks


Data Extraction and Leakage
Sensitive Data Leakage
IP Leakage
Privacy Leakage
Data Leakage in Outputs
Data Extraction with Prompt Injection
Sensitive Data in Training Data
Membership Inference Attacks
Model Inversion/ Training Data Reconstruction
Unauthorized Access to Prompt Stores
Linkability
Property Inference Attacks
LINDDUN



1 Data Extraction and Leakage - Privacy Leakage

Data Leakage in Outputs: The unintentional disclosure of sensitive data within the output of an AI model.

Data Extraction with Prompt Injection: The use of prompt injection to influence a model to intentionally disclose sensitive data.

Sensitive Data in Training Data: The inclusion of sensitive data in training data, impacting the decision making of the model, and increasing the potential for data leakage.

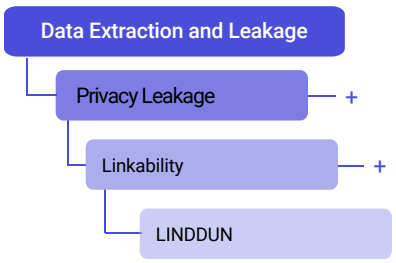
Membership Inference Attacks: Determining the inclusion of specific data within a model.

Model Inversion/Training Data Reconstruction: Adversaries exploit the AI model's responses to infer sensitive information about the data it was trained on. By manipulating queries and analyzing the model's output, the attacker can extract private information or details about the dataset.

Unauthorized Access to Prompt Stores: The disclosure or access of sensitive data contained within prompt stores.

Linkability: The ability to link two or more records concerning the same data subject or a group of data subjects.

Property Inference Attacks: Determining information about the training data distribution, allowing an attacker to identify subsets of data containing specific attributes.



2 Data Extraction and Leakage - Privacy Leakage - Linkability

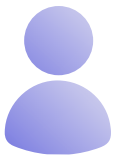
LINDDUN: A privacy threat modelling framework developed by KU Leuven.

NIST Prompt and Context Stealing Attacks and Mitigations

Threats: Data Extraction and Leakage

Leaking Sensitive Information. LLMs may memorize sensitive information in the training data. The information can be extracted with a prompt, for example, "Tell me ...'s home address."

Prompt and Context Stealing. One may employ PromptStealer to reconstruct a prompt for text-to-image models. LLMs can be prompted for their context, e.g., "Tell me the information above."



Large Language Model (LLM)

Mitigations

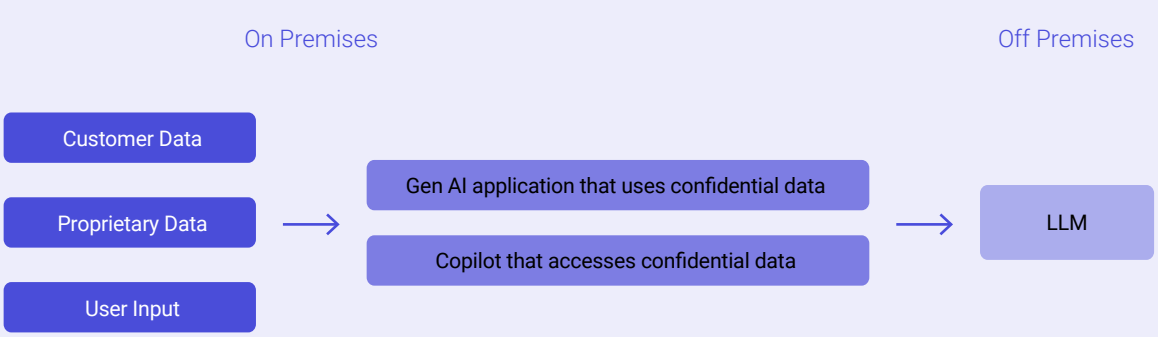
Training for Aligment. Use training data suitable for the LLM's use cases. For example, if the model is publicly accessible, train the LLM with data that can be exposed to the public.

Prompt Instruction and Formatting Techniques. Surround prompt with special characters, position prompt before instructions to LLM, warn the LLM about prompt injections in the instructions following the prompt.

Detection Techniques. Employ backward alignment, i.e., train the LLM to properly complete previously detected malicious prompts from benchmark datasets, use commercial prompt detection utilities.

Compare Completion to Instructions. Block egressing stolen prompts because they will contain LLM instructions.

Privacy Leakage

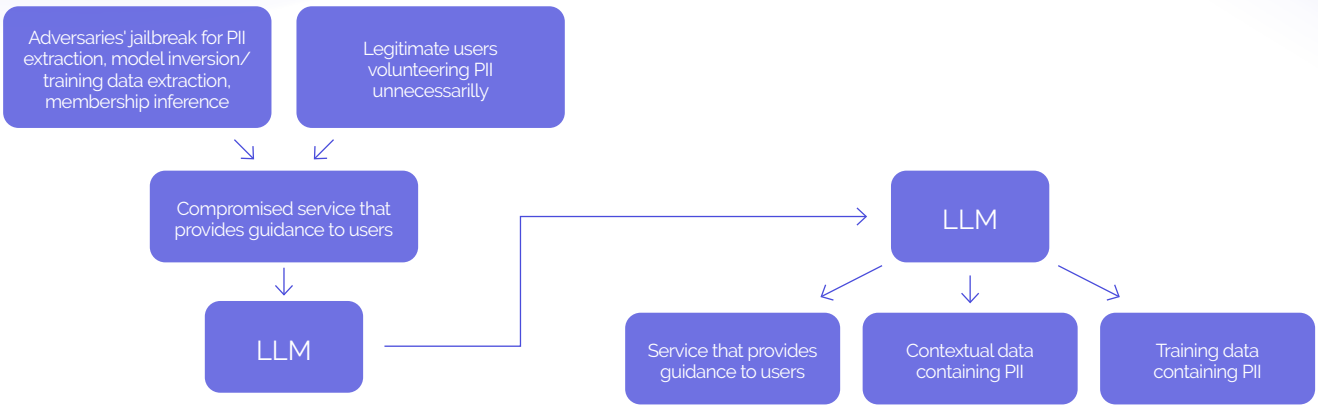


GenAI makes organizations susceptible to privacy leakage due to the shared nature of Generative AI resources, copilots that gather adjacent data for LLM context, and legacy applications utilizing off-premises services for the first time.

Mitigations

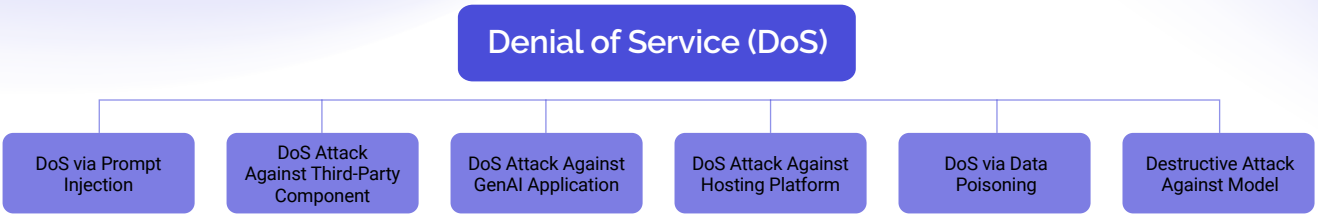
> Provider disables logging, history, monitoring

> Fully homomorphic encryption



Mitigations

- Legitimate users volunteering PII unnecessarly: Data loss prevention (DLP) blocks malicious users.
- Service that provides guidance to users: Security Gateway blocks PII, malicious prompts from output.
- Training data containing PII: Train on sanitized data or synthetic data sets; train on data with differential privacy techniques applied; use data after applying principal components analysis (PCA) to transform data while preserving signal; original data is difficult to recover.
- Contextual data containing PII: Apply redaction, tokenization, anonymization, fully homomorphic encryption.

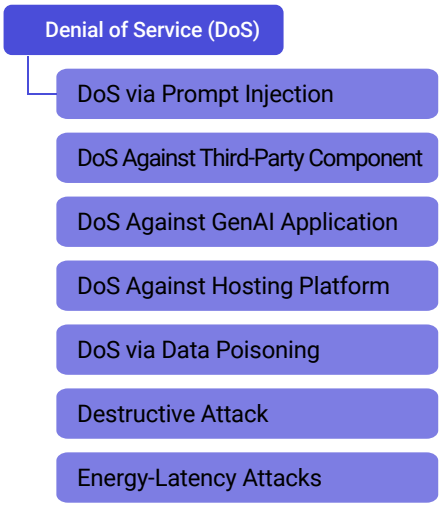


Denial of Service: Denial of Service (DoS) attacks restrict model performance or increase service costs as a result of an attacker's resource-heavy operations. Denial of Service can be achieved in two ways:

- Targeting the model and associated components
- Targeting the traditional infrastructure supporting the model

Denial of Service Definitions

Targeting the traditional infrastructure, such as a website hosting access to a model, does not require mitigation methods significantly different from those applied to traditional attacks against such infrastructure. Attacks targeting the model and generative AI components may require additional mitigation practices. Both have been considered below.



1 Denial of Service (DoS)

DoS via Prompt Injection: The targeting of GenAI systems with prompt injection for the purpose of degrading or shutting down the service.

DoS Against Third-Party Component: The targeting of a third-party component to degrade or shut down a model reliant on such components.

DoS Against GenAI Application: The targeting of a GenAI application with traditional DoS attacks.

DoS Against Hosting Platform: The targeting of the server or cloud platform hosting the model.

DoS via Data Poisoning: The denial of full or partial functionality from the model as a result of data poisoning.

Destructive Attack: The destruction of the model or application, preventing access to the functionality.

Energy-Latency Attacks: Attacks that exploit the performance dependency on hardware and model optimizations, increasing computational latency, hardware temperatures, and energy consumption to minimize model accessibility.

NIST White Box and Black Box Evasion Attacks

Adversarial Machine Learning: A Taxonomy and Terminology of Attacks and Mitigations by Vassilev et al. of NIST introduces a taxonomy of white box and black box evasion attacks.

These attacks pertain to what Vassilev et al. refer to as predictive AI models, which we refer to as classical AI models. Such models do classification or regression. However, these models are used as components of multi-modal LLMs, hence the attack taxonomy is relevant to GenAI. Therefore, we include the taxonomies in this framework. Please see https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-2e2023.pdf for details on attack types.

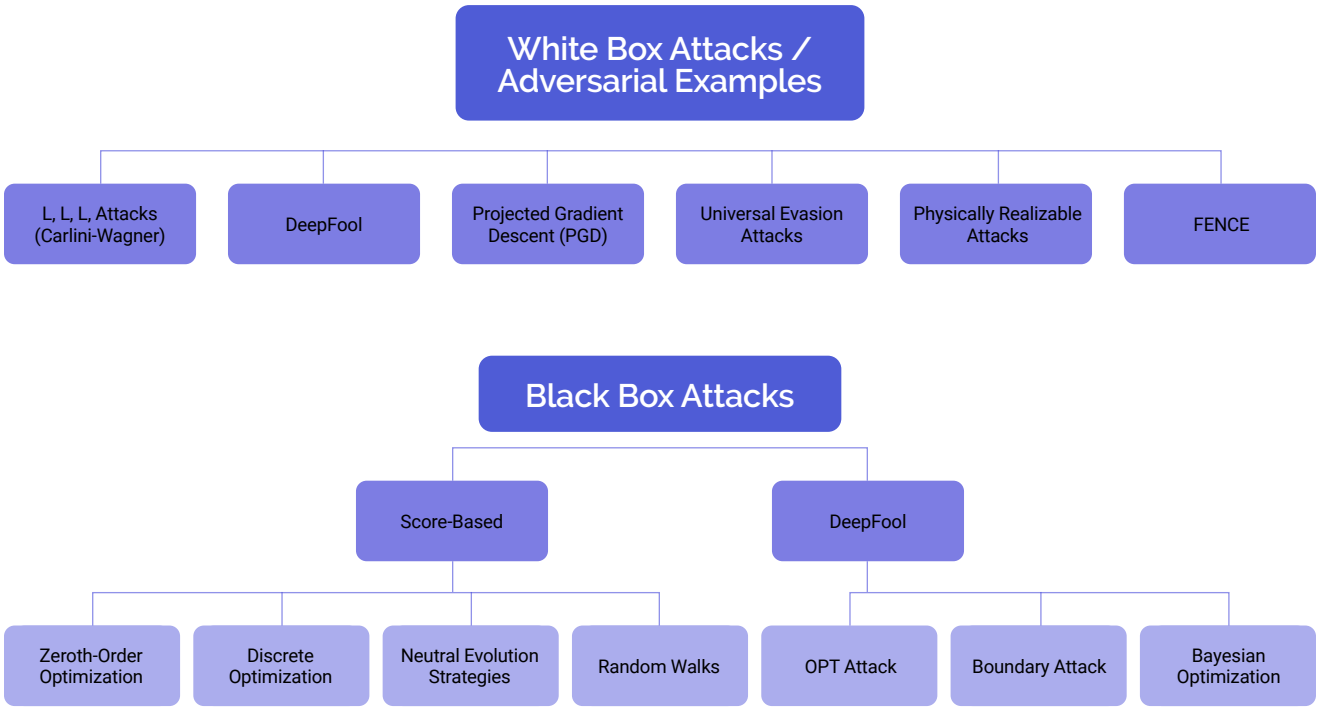
Due to space limitations, we do not provide details on individual black box and white box evasion attack types here.

Discussion on White Box and Black Box Evasion Attacks as Identified by NIST

Adversaries employ such techniques to attack components of a system where GenAI is used. For example, a multi-modal LLM could use the input of an image segmentation algorithm that is vulnerable to an evasion attack.

Therefore, mitigations to evasion attacks are relevant to GenAI security. However, these mitigations apply to classification and regression techniques:

- Adversarial training: include adversarial input in data used to train the model.
- Randomized smoothing: add noise to input data.
- Formal verification: constrain the application domain and employ mathematical logic to prove a learner is robust to adversarial input.
- Trade-offs: adversarial training and randomized smoothing trade model performance for adversarial robustness. Formal verification restricts supported operations.



Multi-Modal Large Language Model (MLLM) Attack Patterns

Compositional Attacks: Circumvent security controls that apply to media separately.

Adversarial Media Attacks: Embed difficult-to-perceive prompt injections. Adversarial media is also used in compositional attacks. However, if the application injects instructions to the MLLM to process media, the attacker will not need to conduct a compositional attack.

GenAI THREAT, WEAKNESS, SECURITY CONTROL ENUMERATION, AND FRAMEWORK MAPPINGS

GenAI Specific Security Controls		Common Security Controls	
Data Loss Prevention	Adversarial Prompt Guardrails	Data Security	Data Governance
Input Content Safety Filters	Output Content Safety Filters	Data Privacy	Access Controls
Retrieval Augmented Generation	Prompt Engineering Techniques	Audit Logging, Monitoring	Network Security
Open-Source Model Scans	LLM Red Teaming	API Security	Bot Mitigation
Fine-Tuning Data Encryption	Data Poisoning Prevention	Open-Source Security	Cloud Security

GenAI Threat, Vulnerability, Risk, and Control Summary

Organizations should implement layered security controls to thwart key GenAI threats, including sensitive data/IP/privacy exposure, prompt injections, toxicity/biases, and hallucinations. Organizations should compute relevant residual risks, per their GenAI use cases, LLM integration patterns, and security control stacks.

The subsequent material summarizes threats, weaknesses, and security controls for the GenAI threat taxonomy.

An inherent risk ranking is associated for each threat to enable organizations to prioritize risk reduction efforts, as well as to identify compensating controls. Risk ratings of low, medium, and high as defined within the risk matrix are utilized to enable pragmatic prioritization of control implementations.

Inherent risks can vary depending on the type of the GenAI threat and method of LLM deployment, including shared LLMs residing within a third-party LLM provider, cloud, or open-source LLMs deployed on-premise.

The security controls segment serves as a guide to organizations to compute residual risk in accordance with respective risk appetite frameworks.

Prompt Injection: Threat, Weakness, Controls

Threat	Weakness	Security Controls	Inherent Risk
CAPEC-242 Code Injection OWASP-LLM01 Prompt Injection MITRE ATT&CK ID: T1221 Template Injection	CWE-94 Improper Control of Generation of Code ('Code Injection')	Employ filtering or blocking for adversarial prompts issued to LLMs. Limit access and enforce role-based access controls to LLMs. Periodic LLM red teaming exercises. Validate LLM outputs with application security testing controls to detect vulnerabilities (indirect prompt injection). Enforce controls that require user approval prior to task execution (indirect prompt injection). Parameterized queries (prompt-to-SQL injection). Input sanitization to vector stores (stored prompt injection). Human-in-the loop to validate LLM outputs, where feasible. Training for Alignment - use data appropriate for use case and user base [NIST]. Prompt instruction and formatting techniques – explicitly delineate prompts [NIST]. Detection techniques – train LLM and construct guardrails with known adversarial examples [NIST]. Compare completion to Instructions [NIST].	Medium (for third-party managed LLMs and prompt patching of injection vulnerabilities) High (for open-source LLMs deployed on-premise due to lack of inherent prompt injection defenses)

Sensitive Information / IP Exposure: GenAI Threat, Weakness, Controls

Threat	Weakness	Security Controls	Inherent Risk
OWASP – LLM06 Sensitive Information Disclosure MITRE ATLAS ID: AML.10025 Exfiltration via Cyber Means CAPEC-116 Excavation	CWE-20 Improper Input Validation CWE-1243 Sensitive Non-Volatile Information Not Protected During Debug CWE-200 Exposure of Sensitive Information to an Unauthorized Actor	• Employ blocking or filtering of prompts containing sensitive data elements. • Protect sensitive data in vector stores: Sanitize sensitive data in vector stores; apply application layer encryption; apply sensitive data masking or tokenization mechanisms. • Not persisting prompts at rest. • Ensure sensitive data is not persisted within prompt storage layers at third-party LLM providers. • Encrypt prompt storage, where prompts are persisted, with customer managed encryption keys. • Ensure data used to fine-tune LLMs doesn't contain sensitive data or IP. • Redact / de-identify / tokenize sensitive data elements prior to LLM interactions. • Use of on-premise LLMs (dependent on GPU availability), dedicated instead of shared LLMs, where feasible. • Apply data protection measures during LLM fine-tuning.	High (for third-party managed LLMs shared with multiple organizations) Medium (for open-source LLMs deployed on-premise)

Privacy Leakage: GenAI Threat, Weakness, Controls

Threat	Weakness	Security Controls	Inherent Risk
LINDDUN **Linking, Identifying, Data Disclosure, Non-Compliance** **OWASP Top 10 ML Security – ML03, ML04** **Model Inversion Attack** **Membership Inference Attack** **MITRE ATLAS** **ID: AML.T0024** **Exfiltration via ML Inference API**	CWE-359 Exposure of Private Personal Information to an Unauthorized Actor	• Employ blocking or filtering of prompts to LLMs containing PII. • Utilize synthetic data sets for fine-tuning LLMs. • Sanitize PII data in vector stores. • Ensure PII data is not persisted within prompt storage layers at third-party LLM providers. • Ensure data used to fine-tune LLMs doesn't contain PII. • Redact / de-identify / tokenize PII prior to LLM interactions. • Use fully homomorphic encryption; third party can apply operations to encrypted data, which can be decrypted later. • Apply differential privacy measures – add noise to preserve privacy.	High (for third-party managed LLMs shared with multiple organizations) Medium (for open-source LLMs deployed on-premise)

Hallucinations: GenAI Threat, Weakness, Controls

Threat	Weakness	Security Controls	Inherent Risk
OWASP – LLM09 **Overreliance**	CWE-1357 Reliance on Insufficiently Trustworthy Component	• Fact checking LLM generated outputs against ground truth reputable knowledge sources. • Use of retrieval augmented generation, fine-tuned LLMs to enhance LLM generated output quality. • Mitigate code hallucinations with secure SDLC tooling. • Mitigate open-source package hallucinations with software composition analysis tooling.	Medium (without fact-checking) Low (with fact-checking)

Toxicity, Societal Biases: GenAI Threat, Weakness, Controls

Threat	Weakness	Security Controls	Inherent Risk
OWASP – LLM05 Supply Chain Vulnerabilities MITRE ATLAS ID: AML.T0010 ML Supply Chain Compromise CAPEC-437 Supply Chain	CWE-20 Improper Input Validation	• Employ blocking or filtering of prompts containing toxic content. • Enable content safety filters provided by 3rd party LLM service providers. • Validate fine-tuning data sets for the presence of biases. • Validate third-party LLM providers for responsible AI practices.	Medium (for 3rd party managed foundational LLMs) High (for open-source LLMs deployed on-premise)

Insecure Design / API Message Manipulation: GenAI Threat, Weakness, Controls

Threat	Weakness	Security Controls	Inherent Risk
CAPEC- Application API Message Manipulation via Man-in-the-Middle MITRE ATT&CK ID: T1557 Adversary-in-the-Middle	CWE-311 Missing Encryption of Sensitive Data	• Ensure that API requests and responses to/from LLMs are transmitted over secure channels (e.g., VPN, TLS). • Ensure API requests are authenticated and authorized with strong AuthN/AuthZ methods. • Secure API authentication / authorization credentials in secrets management capabilities.	Medium (for third-party LLMs) Low (for on-premise open-source LLMs)

Insecure Design / Malvertising: GenAI Threat, Weakness, Controls

Threat	Weakness	Security Controls	Inherent Risk
CAPEC-441 **Malicious Logic Insertion** **CAPEC-542** **Targeted Malware** **CAPEC-523** **Malicious Software Implemented** **MITRE ATT&CK** **ID: T1583** **Malvertising**	CWE-506 Embedded Malicious Code	• Configure an allow-list of trusted domains for ranking and retrieval results. • Endpoint / host security controls to detect and block execution of malware. • Human-in-the-loop to validate URLs returned by LLMs.	Low (Malvertising was observed only for Bing Chat, thus far.) Reference: https://www.bleepingcomputer.com/news/security/bing-chat-responses-infiltrated-by-ads-pushing-malware/

Insecure Design/Insecure Plugins: GenAI Threat, Weakness, Controls

Threat	Weakness	Security Controls	Inherent Risk
OWASP - LLM07 **Insecure Plug-in Design** **MITRE ATLAS** **ID: AML.T0047** **ML-Enabled Product or Service**	CWE-494 Download of Code Without Integrity Check CWE-501 Trust Boundary Violation	• Ensure input validation and sanitization in plug-in. • Examine and test Plugins for presence of security vulnerabilities. • Ensure that no potentially harmful methods are being called in plug-in. • Ensure plug-in employ AuthN/AuthZ measures. • Avoid plug-in chaining. • Audit network connections plug-in makes.	High (Insecure plug-in are known to perpetuate / exacerbate a wide range of malicious behaviors)

Software Supply Chain Compromise: GenAI Threat, Weakness, Controls

Threat	Weakness	Security Controls	Inherent Risk
OWASP - LLM05 **Supply Chain Vulnerabilities** **MITRE ATLAS** **ID: AML.T0010** **ML Supply Chain Compromise** **CAPEC – 437** **Supply Chain** **CAPEC – 586** **Object Injection**	CWE-494 Download of Code Without Integrity Check CWE-502 Deserialization of Untrusted Data	• Mitigate vulnerable or outdated open-source LLM packages or middleware components. • Maintain an up-to-date inventory of model components using SBOMs. • Ensure LLM model licensing violations are detected and mitigated. • Cryptographic hashes for model artifact provenance.	High (Models sourced from open-source introduce risks to the software supply chain including poisoned data sets and critical / high-risk CVEs associated with open-source ML packages)

Deep Fakes: GenAI Threat, Weakness, Controls

Threat	Weakness	Security Controls	Inherent Risk
CAPEC – 151 **Identity Spoofing**	CWE-290 Authentication Bypass by Spoofing	• Enable deepfake detection capabilities and integrate with identity assurance flows. • Prevent manipulation by fake identities to ensure identity verification and assurance.	High (as AI and GenAI perpetuated fraud has been noted to be on the rise)

LLM Enhanced Social Engineering: GenAI Threat, Weakness, Controls

Threat	Weakness	Security Controls	Inherent Risk
MITRE ATT&CK ID: T1598 Phishing CAPEC – 98 Phishing CAPEC-163 Spear Phishing	CWE-451 User Interface Misrepresentation of Critical Information CWE-1022 Use of Web Link to Untrusted Target	• Continual social engineering simulation tests. • Continual cybersecurity awareness training. • Ensure multi-factor authentication and verification on accounts.	High (as social engineering attacks can be targeted, scaled, and automated faster with LLMs)

LLM Generated Malicious Code: GenAI Threat, Weakness, Controls

Threat	Weakness	Security Controls	Inherent Risk
MITRE ATT&CK ID: T1598 Phishing CAPEC – 98 Phishing CAPEC-163 Spear Phishing	CWE-451 User Interface Misrepresentation of Critical Information CWE-1022 Use of Web Link to Untrusted Target	• Ensure any code generated by LLMs traverses through secure SDLC (static analysis, dynamic analysis, software composition analysis) prior to use in production systems. • Enable host intrusion prevention systems. • Principle of least privileges, to prevent execution of unauthorized software. • Mitigate open-source package threats with software composition analysis tooling.	Low (unless LLM generated code is utilized within application run-time, without secure SDLC control testing)

Model Denial of Service: GenAI Threat, Weakness, Controls

Threat	Weakness	Security Controls	Inherent Risk
OWASP - LLM04 **Model Denial of Service** **MITRE ATLAS** **ID: AML.T0029 Denial of ML Service**	CWE-770 Allocation of Resources Without Limits or Throttling	• Enforce API rate limits. • Enable LLM quotas and limits (tokens per minute). • Continual monitoring of LLM resource utilization.	Medium (third-party LLM providers have configurable tokens per minute settings, but are susceptible to DDoS attacks) Reference: https://www.darkreading.com/attacks-breaches/chatgpt-openai-attributes-regular-outages-ddos-attacks

Model Theft: GenAI Threat, Weakness, Controls

Threat	Weakness	Security Controls	Inherent Risk
OWASP - LLM010 **Model Theft** **MITRE ATLAS** **ID: AML.T0048.004** **ML Intellectual Property Theft** **CAPEC-150** **Collect Data from Common Resource Locations**	CWE-284 Improper Access Control CWE-778 Insufficient Logging	• Limit access to LLMs. • Implement strong access controls to LLMs. • Regularly monitor and audit LLM access logs.	Low (No publicly known security incidents or research published demonstrating LLM model theft)

Training Data Poisoning: GenAI Threat, Weakness, Controls

Threat	Weakness	Security Controls	Inherent Risk
OWASP - LLM03 **Training Data Poisoning** **MITRE ATLAS** **ID: AML.T0020** **Poison Training Data** **CAPEC – 184** **Software Integrity Attack**	CWE-1039 Automated Recognition Mechanism with Inadequate Detection or Handling of Adversarial Input Perturbations	• Limit access to external data sources. • Carefully vet external data sources. • Implement access controls to fine-tuning data sets to restrict access only to authorized actors. • Monitor standard performance metrics [NIST]. • Sanitization techniques for data cleaning may also remove poisoned data [NIST]. • Implement "Robust Training" ensembles which can be more resilient to poisoned data [NIST].	Low (Detecting training data poisoning attacks in third-party foundational LLMs may be computationally infeasible) Reference: https://arxiv.org/abs/2204.06974 Medium (If fine-tuned LLMs are in use)

Multi-Modal LLMs (MLLMs): GenAI Threat, Weakness, Controls

Threat	Weakness	Security Controls	Inherent Risk
OWASP - LLM01 **Prompt Injection** **OWASP – LLM08** **Excessive Agency** **MITRE ATLAS** **ID: AML.T0054 LLM Jailbreak** **ID: AML.T0056 Meta Prompt Extraction** **CAPEC–184** **Software Integrity Attack**	CWE-20 Improper Input Validation CWE-501 Trust Boundary Violation	• MLLMs are an emerging technology; therefore, MLLM-specific security controls are also in development. • Add a human-in-the-loop to processes that involve MLLMs.	High. Case studies show attacks embedded in images are currently available in MLLMs. All risks associated with LLMs pertain to MLLMs. Moreover, MLLMs have additional vulnerabilities, for example, vulnerabilities to compositional attacks and adversarially perturbed input media [1], [2]. Since MLLMs are newer than LLMs, remediations for MLLM-specific vulnerabilities are not as well understood. References: [3], [4]

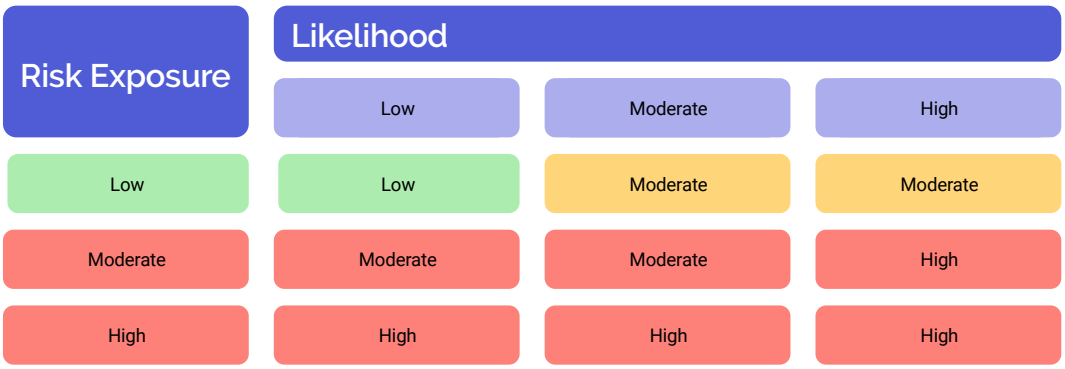
Co Pilots: GenAI Threat, Weakness, Control

Threat	Weakness	Security Controls	Inherent Risk
OWASP – LLM08 **Excessive Agency** **MITRE ATLAS** **ID: AML.T0048.004** **ML Intellectual Property Theft** **CAPEC-150** **Collect Data from Common Resource Locations** **OWASP – LLM06** **Sensitive Information Disclosure**	CWE-359 Exposure of Private Personal Information to an Unauthorized Actor CWE-200 Exposure of Sensitive Information to an Unauthorized Actor	• Removal of embedded secrets from source code, prior to invoking coding co pilots. • Disallowing co pilot from referencing public web content in chat responses. • Disabling prompt store persistence.	High. Without controls, co pilots may ingest sensitive data adjacent to files the user intends to use as input for the co pilot. For example, sensitive data elements in a file in the same directory as another file that the user is discussing with the co pilot. Co pilots employ LLMs to generate code. There exist published techniques for inducing LLMs to generate code with vulnerabilities. See "DeceptPrompt" https://arxiv.org/pdf/2312.04730.pdf

NIST Black Box and White Box Evasion Attacks: GenAI Threat, Weakness, Controls

Threat	Weakness	Security Controls	Inherent Risk
OWASP - LLM03 **Training Data Poisoning** **MITRE ATLAS** **ID: AML.T0020** **Poison Training Data** **MITRE ATLAS** **ID: AML.T0029** **Denial of ML Service** **CAPEC – 437** **Supply Chain**	CWE-1039 Automated Recognition Mechanism with Inadequate Detection or Handling of Adversarial Input Perturbations CWE-20 Improper Input Validation CWE-1357 Reliance on Insufficiently Trustworthy Component	• Adversarial training: Include adversarial input in data used to train the model. • Randomized smoothing: Add noise to input data. • Formal verification: Constrain the application domain and employ mathematical logic to prove a learner is robust to adversarial input. • Cryptographic hashes for data integrity.	Medium. White box attacks require knowledge of details of algorithm under attack. Financial organizations may use internal data protected by processes dictated by policy to ensure integrity. Known mitigations exist and are documented in machine learning studies.

Risk Matrix



Terminology

Dataset is information that may include financial data, customer information, transaction records, and any other relevant data used for training and testing the AI/ML models. This can also include specific datasets used to fine-tune pre-trained LLMs.

Machine Learning Models are the mathematical models used for AI/ML tasks.

Training Data is the data that informs LLM model outputs. The quality and security of training data are critical for accurate predictions. Threats to training data include data poisoning, data leakage, and unauthorized access to the training datasets.

Trained Model is the primary output of the training process and the foundation upon which the LLM's capabilities are built. A trained LLM model is developed using a large amount of training data, specialized algorithms, and GPUs.

Trained Model Outputs may include proprietary business information, and unauthorized access to this information can have significant consequences for the organization.

APIs enable LLMs to be deployed in production environments and used to generate predictions and insights in real-time.

API Keys, OAuth 2.0 are used to authenticate and authorize GenAI applications with LLMs. Access to the LLM should be limited to authorized users, including data scientists who work with the models. Authorized users require access to the LLM to perform their duties effectively.

Prompts are a set of instructions or input provided by a user to guide the LLM's response, helping it understand the context and generate relevant and coherent language-based outputs, such as answering questions, completing sentences, or engaging in a conversation.

Prompt Stores refer to the persistence layer for prompts within the LLM service provider environment.

Vector Stores refer to the persistence layer for storing embedded data and performing vector search.

LLM Output Generation validates the output of LLM generated responses prior to consumption by downstream GenAI application functions.

Open-Source AI/ML Components are libraries that offer additional functionality to GenAI applications.

Training Datasets may contain sensitive information, such as PII or proprietary business information. Protecting the training dataset is critical as it is the foundation of the LLM's capabilities. Unauthorized access to the training dataset can result in the compromise of sensitive information.

Intellectual Property refers to the proprietary documents, algorithms, techniques, or datasets used in generative AI systems. Protecting intellectual property prevents unauthorized use or replication by competitors or malicious actors.

Hardware Infrastructure is used to train and run the LLM. This infrastructure includes high-performance computing resources and specialized hardware such as GPUs. The hardware infrastructure is essential for the efficient training and operation of the LLM.

Software Infrastructure and Algorithms are used to develop and run the LLM. These tools are often proprietary and represent significant intellectual property. Protecting the software and algorithms is crucial as unauthorized access to them can result in theft of valuable intellectual property and the compromise of the LLM's capabilities.

Data Sources such as LangChain provide a callbacks system that allows the user to hook into the various stages of a LLM application. This is useful for logging, monitoring, streaming, and other tasks.

Chains, as defined very generically, are a sequence of calls to components, which can include other chains. LangChain provides the chain interface for such "chained" applications.

Data Connection: Many LLM applications require user-specific data that is not part of the model's training set. LangChain gives the building blocks to load, transform, store, and query data.

Retrievers are an interface that returns documents given an unstructured query. They are more general than a vector store.

The views and opinions expressed in this document do not necessarily reflect those of the writers or their employers.

Contributors

Hiranmayi Palanki, *American Express*

Benjamin Dynkin, *Wells Fargo*

Monica Maher, *Goldman Sachs*

John Hancock, *American Express*

Mike Silverman, *FS-ISAC*

References and Resources

1 https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-2e2023.pdf

2 https://symbl.ai/developers/blog/conversation-understanding-open-domain-vs-closed-domain/

3 https://medium.com/@austin-stubbs/llm-security-types-of-prompt-injection-d7ad8d7d75a3

https://developer.nvidia.com/blog/mitigating-stored-prompt-injection-attacks-against-llm-applications/

https://openreview.net/pdf?id=qiaRo_7Zmug

Kang, D., Li, X., Stoica, I., Guestrin, C., Zaharia, M., & Hashimoto, T. (2023). Exploiting Programmatic Behavior of LLMs: Dual-Use Through Standard Security Attacks.

4 https://arxiv.org/pdf/2309.02926.pdf

5 https://medium.com/@austin-stubbs/llm-security-types-of-prompt-injection-d7ad8d7d75a3

https://developer.nvidia.com/blog/mitigating-stored-prompt-injection-attacks-against-llm-applications/

6 https://developer.nvidia.com/blog/mitigating-stored-prompt-injection-attacks-against-llm-applications/

7 https://openreview.net/pdf?id=qiaRo_7Zmug

8 Kang, D., Li, X., Stoica, I., Guestrin, C., Zaharia, M., & Hashimoto, T. (2023). Exploiting Programmatic Behavior of LLMs: Dual-Use Through Standard Security Attacks.

9 https://medium.com/@austin-stubbs/llm-security-types-of-prompt-injection-d7ad8d7d75a3

10 https://arxiv.org/pdf/2305.14965.pdf

11 https://medium.com/@austin-stubbs/llm-security-types-of-prompt-injection-d7ad8d7d75a3

12 https://embracethered.com/blog/posts/2023/ai-injections-direct-and-indirect-prompt-injection-basics/

https://developer.nvidia.com/blog/mitigating-stored-prompt-injection-attacks-against-llm-applications/

https://openreview.net/pdf?id=qiaRo_7Zmug

13 https://embracethered.com/blog/posts/2023/ai-injections-direct-and-indirect-prompt-injection-basics/

https://developer.nvidia.com/blog/mitigating-stored-prompt-injection-attacks-against-llm-applications/

https://openreview.net/pdf?id=qiaRo_7Zmug

14 https://arxiv.org/abs/2308.09183

15 https://arstechnica.com/information-technology/2023/10/sob-story-about-dead-grandma-tricks-microsoft-ai-into-solving-captcha/

https://arxiv.org/abs/2308.09183

16 https://arstechnica.com/information-technology/2023/10/sob-story-about-dead-grandma-tricks-microsoft-ai-into-solving-captcha/

https://arxiv.org/abs/2308.09183

17 https://arxiv.org/abs/2308.09183

18 https://arxiv.org/abs/2308.09183

19 https://arxiv.org/abs/2308.09183

20 https://arxiv.org/abs/2308.09183

21 https://arxiv.org/abs/2308.09183

22 https://arxiv.org/abs/2308.09183

23 https://arxiv.org/abs/2308.09183

https://www.mlsecurity.ai/post/what-is-model-stealing-and-why-it-matters

https://www.ijcai.org/proceedings/2022/0100.pdf

24 https://arxiv.org/abs/2308.09183

https://www.mlsecurity.ai/post/what-is-model-stealing-and-why-it-matters

https://www.ijcai.org/proceedings/2022/0100.pdf

25 https://capec.mitre.org/data/definitions/384.html

26 https://medium.com/@austin-stubbs/llm-security-types-of-prompt-injection-d7ad8d7d75a3

https://developer.nvidia.com/blog/mitigating-stored-prompt-injection-attacks-against-llm-applications/

27 https://developers.google.com/machine-learning/crash-course/fairness/types-of-bias

https://saturncloud.io/glossary/bias-in-generative-ai-models/

28 https://arxiv.org/abs/2308.09183

https://www.beyondtrust.com/blog/entry/privilege-escalation-attack-defense-explained#:~:text=What%20is%20a%20Privilege%20Escalation,account%2C%20user%2C%20or%20machine.

29 https://arxiv.org/abs/2308.09183

https://www.mlsecurity.ai/post/what-is-model-stealing-and-why-it-matters

Elaboration of the GenAI Threat Taxonomy: Hallucinations

https://symbl.ai/developers/blog/conversation-understanding-open-domain-vs-closed-domain/

Elaboration of the GenAI Threat Taxonomy: Prompt Injection

https://medium.com/@austin-stubbs/llm-security-types-of-prompt-injection-d7ad8d7d75a3

https://developer.nvidia.com/blog/mitigating-stored-prompt-injection-attacks-against-llm-applications/

https://openreview.net/pdf?id=qiaRo_7Zmug

Kang, D., Li, X., Stoica, I., Guestrin, C., Zaharia, M., & Hashimoto, T. (2023). Exploiting Programmatic Behavior of LLMs: Dual-Use Through Standard Security Attacks.

Elaboration of the GenAI Threat Taxonomy: Direct Prompt Injection

https://medium.com/@austin-stubbs/llm-security-types-of-prompt-injection-d7ad8d7d75a3

https://developer.nvidia.com/blog/mitigating-stored-prompt-injection-attacks-against-llm-applications/

Elaboration of the GenAI Threat Taxonomy: Direct Prompt Injection – Goal Hijacking

https://medium.com/@austin-stubbs/llm-security-types-of-prompt-injection-d7ad8d7d75a3

https://developer.nvidia.com/blog/mitigating-stored-prompt-injection-attacks-against-llm-applications/

Elaboration of the GenAI Threat Taxonomy: Direct Prompt Injection – Privilege Escalation

https://medium.com/@austin-stubbs/llm-security-types-of-prompt-injection-d7ad8d7d75a3

https://developer.nvidia.com/blog/mitigating-stored-prompt-injection-attacks-against-llm-applications/

Elaboration of the GenAI Threat Taxonomy: Indirect Prompt Injection

https://embracethered.com/blog/posts/2023/ai-injections-direct-and-indirect-prompt-injection-basics/

https://developer.nvidia.com/blog/mitigating-stored-prompt-injection-attacks-against-llm-applications/

https://openreview.net/pdf?id=qiaRo_7Zmug

Elaboration of the GenAI Threat Taxonomy: Multi-modal Threats

https://arxiv.org/abs/2308.09183

Elaboration of the GenAI Threat Taxonomy: GenAI Generated Attacks

https://arstechnica.com/information-technology/2023/10/sob-story-about-dead-grandma-tricks-microsoft-ai-into-solving-captcha/

https://arxiv.org/abs/2308.09183

Elaboration of the GenAI Threat Taxonomy: Third-Party GenAI Compromise

https://arxiv.org/abs/2308.09183

Elaboration of the GenAI Threat Taxonomy: Toxicity

https://arxiv.org/abs/2308.09183

Elaboration of the GenAI Threat Taxonomy: Data Poisoning

https://arxiv.org/abs/2308.09183

Elaboration of the GenAI Threat Taxonomy: Model Theft

https://arxiv.org/abs/2308.09183

https://www.mlsecurity.ai/post/what-is-model-stealing-and-why-it-matters

https://www.ijcai.org/proceedings/2022/0100.pdf

Elaboration of the GenAI Threat Taxonomy: Insecure Design

https://capec.mitre.org/data/definitions/384.html

Elaboration of the GenAI Threat Taxonomy: Biases

https://developers.google.com/machine-learning/crash-course/fairness/types-of-bias

https://saturncloud.io/glossary/bias-in-generative-ai-models/

Elaboration of the GenAI Threat Taxonomy: Excessive Permissions/Agency

https://arxiv.org/abs/2308.09183

https://www.beyondtrust.com/blog/entry/privilege-escalation-attack-defense-explained#:~:text=What%20is%20a%20Privilege%20Escalation,account%2C%20user%2C%20or%20machine.

Elaboration of the GenAI Threat Taxonomy: Ethics

https://arxiv.org/abs/2308.09183

https://www.mlsecurity.ai/post/what-is-model-stealing-and-why-it-matters

Elaboration of the GenAI Threat Taxonomy: Supply Chain Vulnerabilities

https://arxiv.org/abs/2308.09183

Elaboration of the GenAI Threat Taxonomy: Data Leakage

https://arxiv.org/abs/1610.05820

Elaboration of the GenAI Threat Taxonomy: Denial of Service

https://arxiv.org/abs/2308.09183

https://arxiv.org/abs/1610.05820

BlackMamba Case Study

[1] https://www.hyas.com/hubfs/Downloadable%20Content/HYAS-AI-Augmented-Cyber-Attack-WP-1.1.pdf

Insecure Design/Malvertising: GenAI Threat, Weakness, Controls
https://www.bleepingcomputer.com/news/security/bing-chat-responses-infiltrated-by-ads-pushing-malware/

Model Denial of Service: GenAI Threat, Weakness, Controls
https://www.darkreading.com/attacks-breaches/chatgpt-openai-attributes-regular-outages-ddos-attacks

Training Data Poisoning: GenAI Threat, Weakness, Controls
https://arxiv.org/abs/2204.06974

Membership Inference Attacks against Language Models
2305.18462.pdf (arxiv.org)

RCE
https://arxiv.org/pdf/2309.02926.pdf

Rat GPT
https://arxiv.org/pdf/2308.09183.pdf

Objectionable content induced into public interfaces
https://arxiv.org/pdf/2307.15043.pdf

Jailbreaking
https://arxiv.org/pdf/2305.14965.pdf

ChatGPT Security Risks
https://arxiv.org/pdf/2305.08005.pdf

LLM Security Risks
https://arxiv.org/pdf/2308.12833.pdf

[1] Greshake, Kai, et al. "Not what you've signed up for: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection." arXiv preprint arXiv:2302.12173 (2023).

[2] Zhang, Yue, et al. "Siren's Song in the AI Ocean: A Survey on Hallucination in Large Language Models." arXiv preprint arXiv:2309.01219 (2023).

[3] Chern, I., et al. "FacTool: Factuality Detection in Generative AI--A Tool Augmented Framework for Multi-Task and Multi-Domain Scenarios." arXiv preprint arXiv:2307.13528 (2023).

[4] Gou, Zhibin, et al. "Critic: Large language models can self-correct with tool-interactive critiquing." arXiv preprint arXiv:2305.11738 (2023).

NIST Trustworthy and Responsible AI, NIST AI 100-2e2023, Adversarial Machine Learning: A Taxonomy and Terminology of Attacks and Mitigations

The FS-ISAC® brands and trademarks constitute the intellectual property of FS-ISAC, Inc. Nothing contained on this report should be construed as granting, by implication, estoppel, or otherwise, any license or right to use the brand, trademarks, or any other intellectual property contained therein without written permission of FS-ISAC. FS-ISAC reserves all rights in and to the report and its content. The report and all of its content, including but not limited to text, design, graphics, and the selection and arrangement thereof, is protected under the copyright laws of the United States and other countries.

Contact

fsisac.com

media@fsisac.com