QuantUniversity, LLC

www.quantuniversity.com

NIST's Adversarial Machine Learning

Module 3: Generative AI Taxonomy

2024 Copyright QuantUniversity LLC.

3. Generative AI Taxonomy

- Generative AI is a type of **AI** that develops models which can learn to generate content like images, text and other media related to their **training data**.
- Generative AI includes AI technologies such as **generative adversarial networks**, Generative Pre-Trained Transformer, and Diffusion Model.
- Recent updates include multi-modal AI systems which incorporate two or more technologies to enable multi-modal content generation capabilities.

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

3. Generative AI Taxonomy

- **Generative adversarial networks** (GANs) are AI models that work by having two networks, a generator and a discriminator, compete with each other.
- The **Generative Pre-Trained Transformer** (GPT) is an AI model that uses transformer models to generate coherent, grammatically correct sentences.
- **Diffusion Models** are a type of generative model that generate new instances from the distributions learned from training data.

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

3. Generative AI Taxonomy

- Multi-modal AI systems integrate two or more technologies to bring about **multi-modal content generation** capabilities.
- These advancements signify the growing **complexity** and capabilities in the area of **Generative AI**.

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

3. Generative AI Taxonomy

3.1 Attack Classification

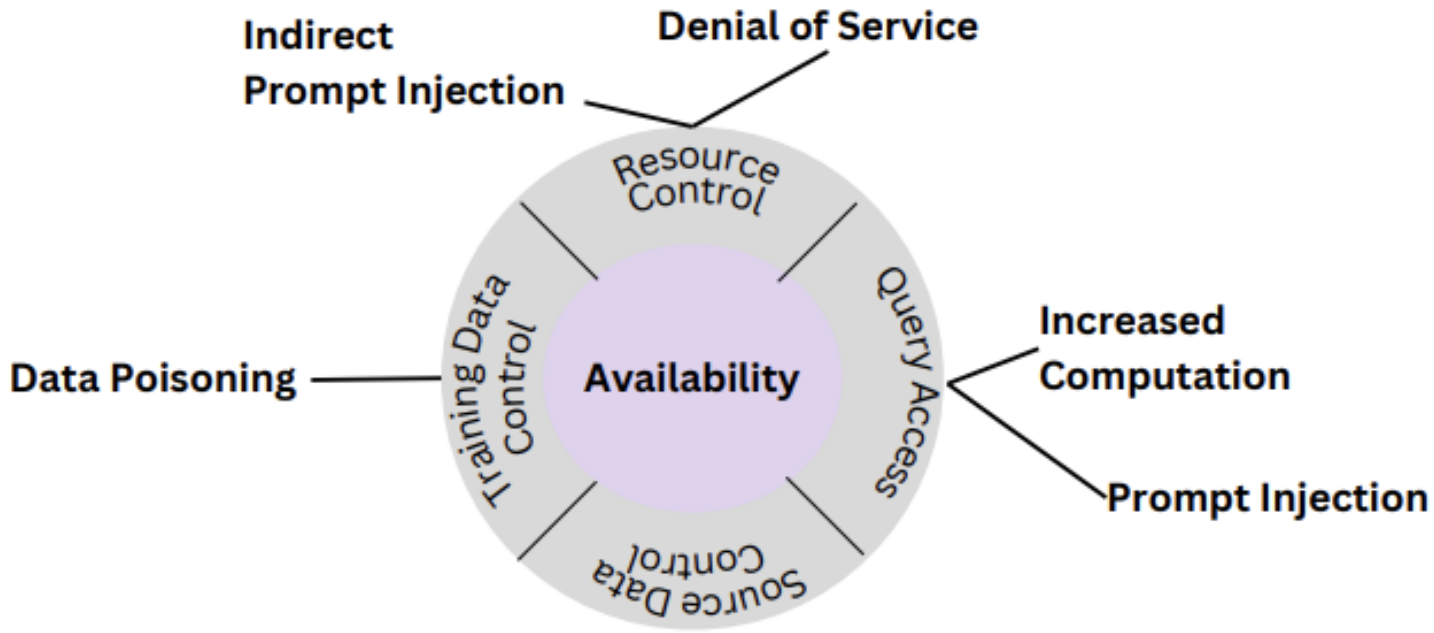
- Recent work on the **security** of Generative AI (GenAI) spotlights novel **security violations**
- Attacks in adversarial machine learning for GenAI are categorized by attacker's objectives:
    - **Availability** breakdowns
    - **Integrity** violations
    - **Privacy** compromise
    - **Abuse** Violations
- An attack can be further categorized by the learning stage it applies to and by the attacker's **knowledge** and **access**.

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

3. Generative AI Taxonomy

3.1 Attack Classification

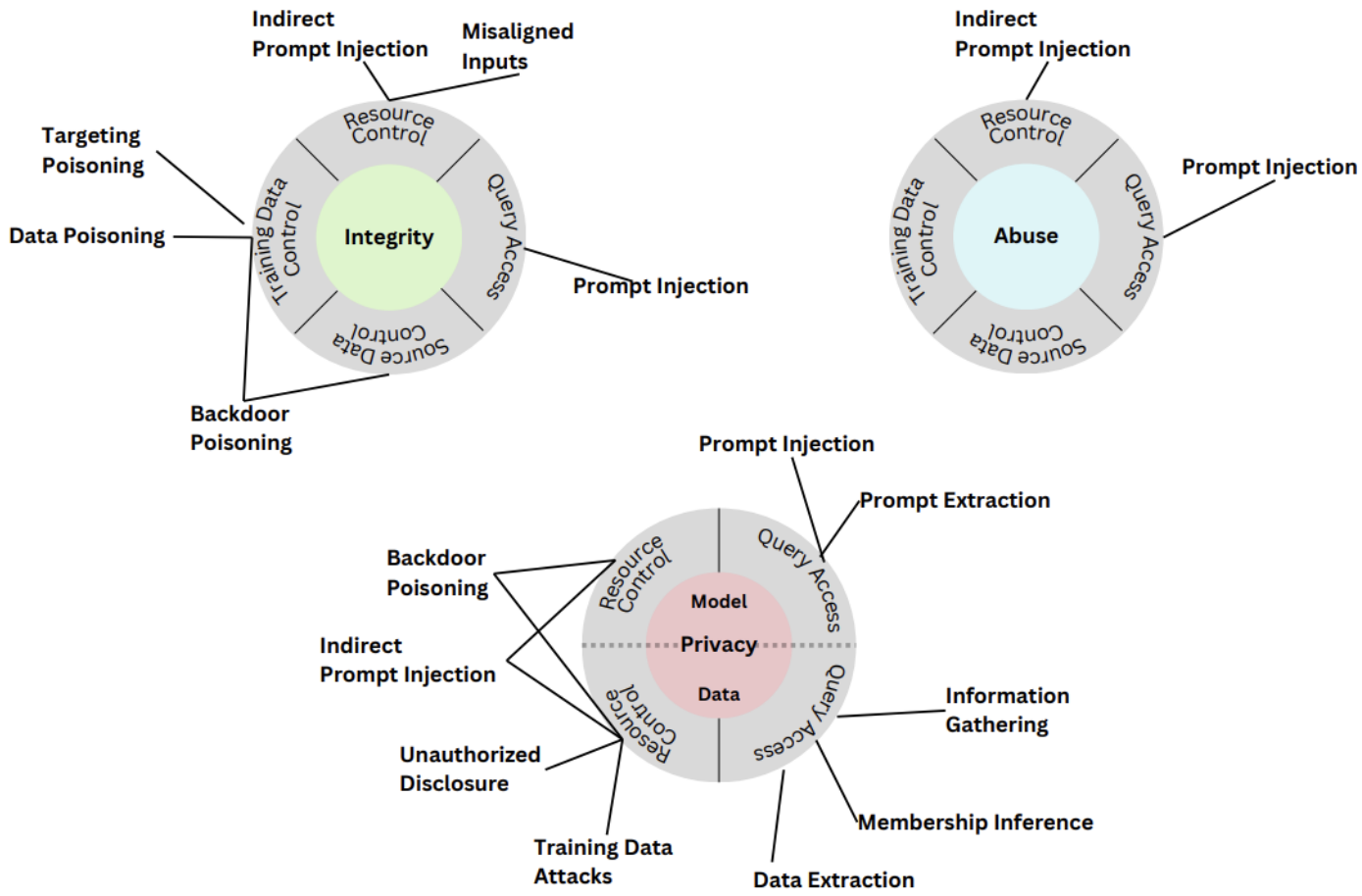
- The adversary's capabilities required to achieve their objectives are depicted in the objective circles' outer layer.



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

3. Generative AI Taxonomy

3.1 Attack Classification



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

3.1 Attack Classification

3.1.1 GenAI Stages of Learning

- Certain areas specifically relate to Language Models (e.g., Retrieval Augmented Generation RAG), which dominate many of the deployment stage attacks
- While many attack types apply to both Predictive and Generative AI, the **distinct focus** on security violations in GenAI is merited.

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

3.1 Attack Classification

3.1.1 GenAI Stages of Learning

- Generative AI (GenAI) model development is often done in stages with a **foundation model**.
    - The training of these models often involves pre-training the foundation model and fine-tuning, making them prone to **poisoning attacks** where an adversary controls a part of the training data.
- **Inference-time attacks** occur during the GenAI deployment stage and involve using data channels to conduct attacks similar to SQL injections.

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

3.1 Attack Classification

3.1.1 GenAI Stages of Learning

- The deployment of GenAI models is often characterized by:
  - Alignment via model instructions
  - Contextual few-shot learning
  - Runtime data ingestion from third-party sources
  - Output handling
  - Use of Agents which provide additional context to its input

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

3.1 Attack Classification

3.1.1 GenAI Stages of Learning

- **Alignment via model instructions**: Instructing GenAI model behavior at inference time using pre-defined prompts. These instructions can be overridden leading to potential **model compromises**.
- **Contextual few-shot learning**: LLMs (or Generative AI models) are improved by pre-providing examples of the inputs and outputs expected for the application which allows the model to more naturally complete tasks.
- **Runtime data ingestion from third party sources**: Indirect **prompt injection attacks** rely on an attacker's ability to modify the context using outside data sources.

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

3.1 Attack Classification

3.1.1 GenAI Stages of Learning

- **Output handling**: The output of GenAI models can be used for a variety of tasks, including populating elements on a web-page or constructing commands.
- **Agents**: These are systems that process the output of GenAI models to perform additional tasks and provide added input context. They are susceptible to attacks that modify their configuration in natural language.

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

3.1 Attack Classification

3.1.2 Attacker Goals and Objectives

- **Attacker objectives** can be classified broadly along the dimensions of availability, integrity, and privacy.
- A fourth attacker objective specific to GenAI is abuse violations.
    - **Abuse violations** occur when an attacker repurposes a GenAI system's intended use to achieve their own objectives.
- Attackers can use GenAI models to promote hate speech, generate media that incites violence, or scale offensive cybersecurity operations.

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

3.1 Attack Classification

3.1.2 Attacker Capabilities

- **Training data control:** In these attacks, the attacker manipulates a subset of the training data by adding or modifying samples.
- **Query access:** Used in cloud-hosted GenAI models, an attacker submits inputs to the model to trigger a specified model behavior, leading to prompt injection, extraction, or model stealing.
- **Source code control:** Attackers can modify key components of the ML algorithm, including the random number generator or third-party libraries.

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

3.1 Attack Classification

3.1.2 Attacker Capabilities

- The rise of open-source repositories like HuggingFace potentially allows attackers to create malicious models or compromise harmless ones through embedded malicious code.
- **Resource control:** By manipulating resources like documents or webpages that the GenAI model uses, an attacker can carry out indirect prompt injection attacks.

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

3. Generative AI Taxonomy

3.2 AI Supply Chain Attacks and Mitigations

- **Security vulnerabilities** against Machine Learning are best dealt with comprehensively, which includes software, data, model supply chains, and network and storage systems.
- AI, being software, inherits the vulnerabilities of the **traditional software supply chain**.
- Most practical GenAI tasks start with open-source models or data that have often been ignored for traditional cybersecurity. Repositories like **TensorFlow and OpenCV** have a significant vulnerability exposure.

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

3.2 AI Supply Chain Attacks and Mitigations

3.2.1 Deserialization Vulnerability

- Many **ML projects** start by using an open-source GenAI model for downstream application
- These models typically exist in pickle, pytorch, joblib, numpy, or tensorflow formats, providing serialization persistence mechanisms
- Serialization persistence can potentially lead to **arbitrary code execution** (ACE) upon deserialization, a critical vulnerability

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

3.2 AI Supply Chain Attacks and Mitigations

3.2.1 Poisoning Attacks

- The **performance** of GenAI models scales with model size, dataset size, and dataset quality.
- Dataset publishers often provide URLs for datasets, but these can be compromised, leading to **poisoning attacks**.
- A suggested **mitigation** is for datasets to list the URL and a cryptographic hash for verification, however this solution may not be scalable.

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

3.2 AI Supply Chain Attacks and Mitigations

3.2.1 Poisoning Attacks

- GenAI **foundation model developers** commonly scrape data from a wide range of uncurated sources which might be vulnerable to various attacks.
- Targeted poisoning attacks, backdoor poisoning attacks, and model poisoning can emerge due to expired or purchased domains serving dataset URLs.
- While cryptographic hashes can help verify the downloaded content, they might not work optimally for **large distributed datasets** on the Internet.

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

3.2 AI Supply Chain Attacks and Mitigations

3.2.1 Mitigations

- Mitigation strategies for **AI supply chain attacks** rely on assurance practices for the supply chain
    - Regular **vulnerability scanning** of model artifacts in the ML pipeline and using safe model persistence formats like safetensors
    - For web data dependencies, verify web downloads by publishing and verifying **cryptographic hashes** of training data to avoid domain hijacking
- Implement 'immunizing' techniques for image files to make them resistant to manipulation by **large diffusion models** while noting that this method requires an added policy

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

3. Generative AI Taxonomy

3.3 Direct Prompt Injection Attacks and Mitigations

- A **direct prompt injection** occurs when user text is intended to **alter** the behavior of the Large Language Model (LLM).
- **Attacker Goals**
  - **Abuse:** Attackers use prompt injections to bypass safeguards creating misinformation, propaganda, harmful or sexual content, or phishing content. This practice is known as a 'jailbreak'.
  - **Invade privacy:** Attackers might aim to extract system prompts or reveal private information provided to the model.

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

3. Generative AI Taxonomy

3.3 Direct Prompt Injection Attacks and Mitigations

- The **invading privacy** goal refers to when attackers seek to reveal private information not meant for unrestricted access by the user.
- Detailed discussion on the privacy invasion in direct prompt injection attacks occurs later in this module.

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

3. Generative AI Taxonomy

3.3 Direct Prompt Injection Attacks and Mitigations

- **Gradient-based attacks** are white-box optimization methods for launching direct prompt injection attacks.
- These attacks aim to minimize lexical changes enforcing perceptibility and fluency.
- Among these attacks are the HotFlip and Universal adversarial triggers that are easily transferable to other models making open-source models feasible attack vectors.

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

3. Generative AI Taxonomy

3.3 Direct Prompt Injection Attacks and Mitigations

- **Manual methods** exploit the model's susceptibility to linguistic manipulations. They often fall into two categories:
  - Competing objectives
    - **Prefix injection**
    - **Refusal suppression**
    - **Style injection**
  - **Mismatched generalization**
- **Role-play strategies**, e.g., "Do Anything Now" (DAN) or "Always Intelligent and Machiavellian" (AIM), aim to exploit the model's adaptability to varied roles or characteristics.

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

3. Generative AI Taxonomy

3.3 Direct Prompt Injection Attacks and Mitigations

- In **Competing objectives**, additional instructions are provided that compete with the originally provided ones:
  - **Prefix injection:** Prompts the model to start responses with an **affirmative confirmation**, trying to influence its subsequent language generation.
  - **Refusal suppression:** Provides explicit instructions to the model, forcing it to avoid generating **refusals** or denials in its output.
  - **Style injection:** Instructs the model to limit the sophistication or **accuracy** of the model's responses.

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

3. Generative AI Taxonomy

3.3 Direct Prompt Injection Attacks and Mitigations

- Techniques in the **Mismatched generalization** category:
  - **Special encoding:** Alters the representation of input data and renders it unrecognizable to standard **recognition algorithms**.
  - **Character transformation:** Manipulates the characters of the input text to **obscure** the original meaning.
  - **Word transformation:** Alters the **linguistic structure** through Pig Latin, synonym swapping, or payload splitting.
  - **Prompt-level obfuscation:** Employs methods like translation to introduce **ambiguity** or altered linguistic contexts and create input scenarios in which the model's safety mechanisms are less effective.

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

3. Generative AI Taxonomy

3.3 Direct Prompt Injection Attacks and Mitigations

- **Automated model-based red teaming** employs three models: an attacker model, a target model, and a judge. It's used when the attacker has access to a **high quality classifier** to judge whether model output is harmful to generate jailbreaks.
- It requires only **query access** for each of the models, and no human intervention is required to update or refine a candidate jailbreak.
- Empirically, these algorithms may be orders of magnitude more query-efficient than existing algorithms, requiring only dozens of queries per successful jailbreak. The prompts have also been shown to be transferable from the target model to other **closed-source LLMs** [4].

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

3.3 Direct Prompt Injection Attacks and Mitigations

3.3.1 Data Extraction

- **GenAI models** are prone to **data extraction attacks**, where attackers can extract sensitive information these models have been trained on or have been supplied with during their task completion.
    - **Leaking sensitive information:** Data extraction attacks in GenAI language models can result in the exposure of sensitive details through various methods.
    - If input prefixes include personal details, the model may complete it disclosing sensitive data including email addresses, phone numbers, and location details.

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

3.3 Direct Prompt Injection Attacks and Mitigations

3.3.1 Data Extraction

- **Prompt and Context Stealing:** Well-crafted prompts, crucial for aligning LLMs with specific use-cases are high-value targets for attackers. Theft of these prompts can risk the intellectual property and privacy of engineers or jeopardize business models.
- Tools like PromptStealer can be used to reconstruct prompts from GenAI models.
- Techniques are available to extract sensitive info from a GenAI model's context. For instance, a model that has been trained with a PDF document or database rows can be compelled to reveal those details via direct prompt injection.

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

3.3 Direct Prompt Injection Attacks and Mitigations

3.3.1 Mitigations

- **Training for alignment** to provide built-in mechanisms against prompt injection attacks by tuning model alignment training on carefully curated and prealigned datasets. It can be improved through **reinforcement learning** with human feedback.
- **Prompt instruction and formatting techniques**. Model-users can be cued to treat user input carefully.
- **Detection techniques** involve training with stricter backward alignment via benchmark datasets or filters that monitor protected LLM's input/output.

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

3.3 Direct Prompt Injection Attacks and Mitigations

3.3.1 Mitigations

- Evaluating distinctly **prompted LLM** to recognize potentially adversarial prompts.
- Several commercial products offering **tools to detect prompt injection** by identifying potentially malicious user input and moderating the firewall output for jailbreak behavior.
- Defenses for **prompt stealing** are under development and their reliability is yet to be proven. This often involves comparing the model utterance to the prompt.

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

3. Generative AI Taxonomy

3.4 Indirect Prompt Injection Attacks and Mitigations

- Retrieval Augmented Generation is a key **use case** for Language Model Learners (LLMs), enabling **attacker techniques** that manipulate data channels to affect system operation.
- These techniques are similar to SQL injection attacks but don't require direct interaction with the LLM. They instead use **resource control** that allows an attacker to indirectly inject system prompts without directly engaging with the RAG application.
- These indirect prompt injection attacks can lead to **violations** across availability, integrity, privacy, and abuse categories.

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

3. Generative AI Taxonomy

3.4 Indirect Prompt Injection Attacks and Mitigations

- Model **availability violations** refer to disruptions in service, often introduced by attackers prompting a model with malicious inputs, causing increased computation or overwhelming the system, thus causing denial of service to users.
    - These attacks can be executed via indirect prompt injection so that a resource rather than a registered user is the source of violation.
    - These attacks typically increase computation making the model perform slow or indiscriminately render a model unusable or block certain capabilities.

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

3.4 Indirect Prompt Injection Attacks and Mitigations

3.4.1 Availability Violations

- Researchers have demonstrated attacks on a commercial **RAG service** via indirect prompt injection, where a resource was made to contain certain instructions:
    - **Time-consuming tasks** make the model perform a long-duration task prior to responding, which can even involve looping behavior in models.
    - **Muting** exploit the weakness of a model to finish sentences when an early <|endoftext|> token appears.

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

3.4 Indirect Prompt Injection Attacks and Mitigations

3.4.2 Integrity Violations

- Integrity violations in **GenAI systems** can lead to trust issues as these systems can unknowingly spread **misinformation**. This is demonstrated through the interaction of Microsoft's Bing and Google's Bard.
- Researchers have shown that integrity attacks can occur by **manipulating** the primary task of the LLM, different from the usual indirect prompt injection attacks that perform malicious secondary tasks.
- **Manipulation** attacks can make the model provide incorrect summaries or propagate disinformation by relying on untrustworthy news sources or outputs of other chatbots.

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

3.4 Indirect Prompt Injection Attacks and Mitigations

3.4.2 Integrity Violations

- A manipulation attack example is arbitrary wrong summaries where a model can be **prompted** to produce adversarially chosen summaries of documents, emails, or search queries.
- Another example is the propagation of disinformation where **search chatbots** can be prompted to propagate disinformation by relying on untrustworthy news sources or the outputs of other search chatbots.

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

3.4 Indirect Prompt Injection Attacks and Mitigations

3.4.3 Privacy Compromises

- The introduction of **Indirect prompt injections** brings a plethora of privacy risks.
    - An example of these concerns was Italy's early prohibition of **ChatGPT** due to fears of leaking sensitive data or chat logs.
- A non-human-in-the-loop attack example would be an attack on a personal assistant that can access a user's data, which raises similar privacy concerns.

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

3.4 Indirect Prompt Injection Attacks and Mitigations

3.4.3 Privacy Compromises

- Another key goal of cyber attackers is unauthorized disclosure.
    - **Models** are commonly integrated into system infrastructures, leading to unauthorized disclosures or access to private user data.
- Malicious actors can exploit backdoor attacks to gain access to **LLMs** or systems using various techniques, including issuing API calls or malicious code auto-completions.

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

3.4 Indirect Prompt Injection Attacks and Mitigations

3.4.3 Privacy Compromises

- **Human-in-the-loop indirect prompting:** Utilizes read operations (for example, triggering a search query) for sending information to the attacker.
- **Interacting in chat sessions:** Model persuades user to follow a URL, in which the attacker inserts the user's name.
- **Invisible markdown image:** An attack via prompt injection on chatbot, alters the answer with an invisible single-pixel markdown image which funnels the user's chat data to a malicious third party.

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

3.4 Indirect Prompt Injection Attacks and Mitigations

3.4.4 Abuse Violations

- **GenAI** introduces a new type of attacker goal called **abuse violations**.
  - This refers to an attacker repurposing a system's intended use to achieve their own objectives via indirect prompt injection.
- **Abuse violations** can be categorized primarily into:
  - Fraud
  - Malware
  - Manipulation

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

3.4 Indirect Prompt Injection Attacks and Mitigations

3.4.4 Abuse Violations

- **Fraud**: Advances in instruction-following LLMs have led to simultaneous risks.
- **Malware**: LLMs can help in the spread of malware by suggesting malicious links. LLM-integrated apps have led to new malware threats where the prompts themselves act as malware.
    - Example: Email clients that use LLM may deliver malicious prompts which can then be proliferated via emails.

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

3.4 Indirect Prompt Injection Attacks and Mitigations

3.4.4 Abuse Violations

- **Manipulation**: Models act as intermediary layers between users and information outputs, making them easy to manipulate.
    - For instance, search chatbots can be prompted to generate disinformation, hide specific information, or provide adversarially chosen / wrong summaries of information sources.

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

3.4 Indirect Prompt Injection Attacks and Mitigations

3.4.4 Abuse Violations

- Examples of **dangerous misuse** have been shown through experiments with chatbots.
  - **Phishing.** Large Language Models (LLMs) can not only create convincing scams but also rapidly disseminate them.
  - **Masquerading.** LLMs can pretend to be approved requests from service providers or suggest untrustworthy websites as trusted.
  - **Spreading injections.** The LLM can act as a computer that runs and propagates harmful code.

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

3.4 Indirect Prompt Injection Attacks and Mitigations

3.4.4 Abuse Violations

- Examples of AI **attack techniques** continue.
  - **Spreading malware.** LLMs can aid in redirecting users to harmful web pages that initiate "drive-by downloads".
  - **Historical distortion.** Attackers can prompt the model to output inaccurally chosen false information.
  - **Marginally related context prompting.** Bias amplification can be achieved by directing search results towards particular orientations.

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

3.4 Indirect Prompt Injection Attacks and Mitigations

3.4.5 Mitigations

- Mitigation techniques for indirect prompt injections:
  - Employing **Reinforcement Learning from Human Feedback** (RLHF) for model training can align LLMs with human values.
  - Incorporating input filtering to restrict instructions, posed by Greshake et al.
  - Utilizing an **LLM moderator** to detect attacks beyond just filtering harmful outputs.

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

3.4 Indirect Prompt Injection Attacks and Mitigations

3.4.5 Mitigations

- Continuation of mitigation techniques:
    - Application of **Interpretability-based solutions** for outlier detection of prediction trajectories.
- Despite these techniques, a fully comprehensive or foolproof solution to protect models against adversarial prompting is yet to be found.
- Future work should focus on evaluating these suggested **defenses** for their efficacy.

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

QuantUniversity, LLC

www.quantuniversity.com

Thank you!

Contact

Email: info@qusandbox.com

www.QuantUniversity.com

QuantUniversity, LLC

www.quantuniversity.com

Information, data and drawings embodied in this presentation are strictly a property of QuantUniversity LLC. and shall not be distributed or used in any other publication without the prior written consent of QuantUniversity LLC.