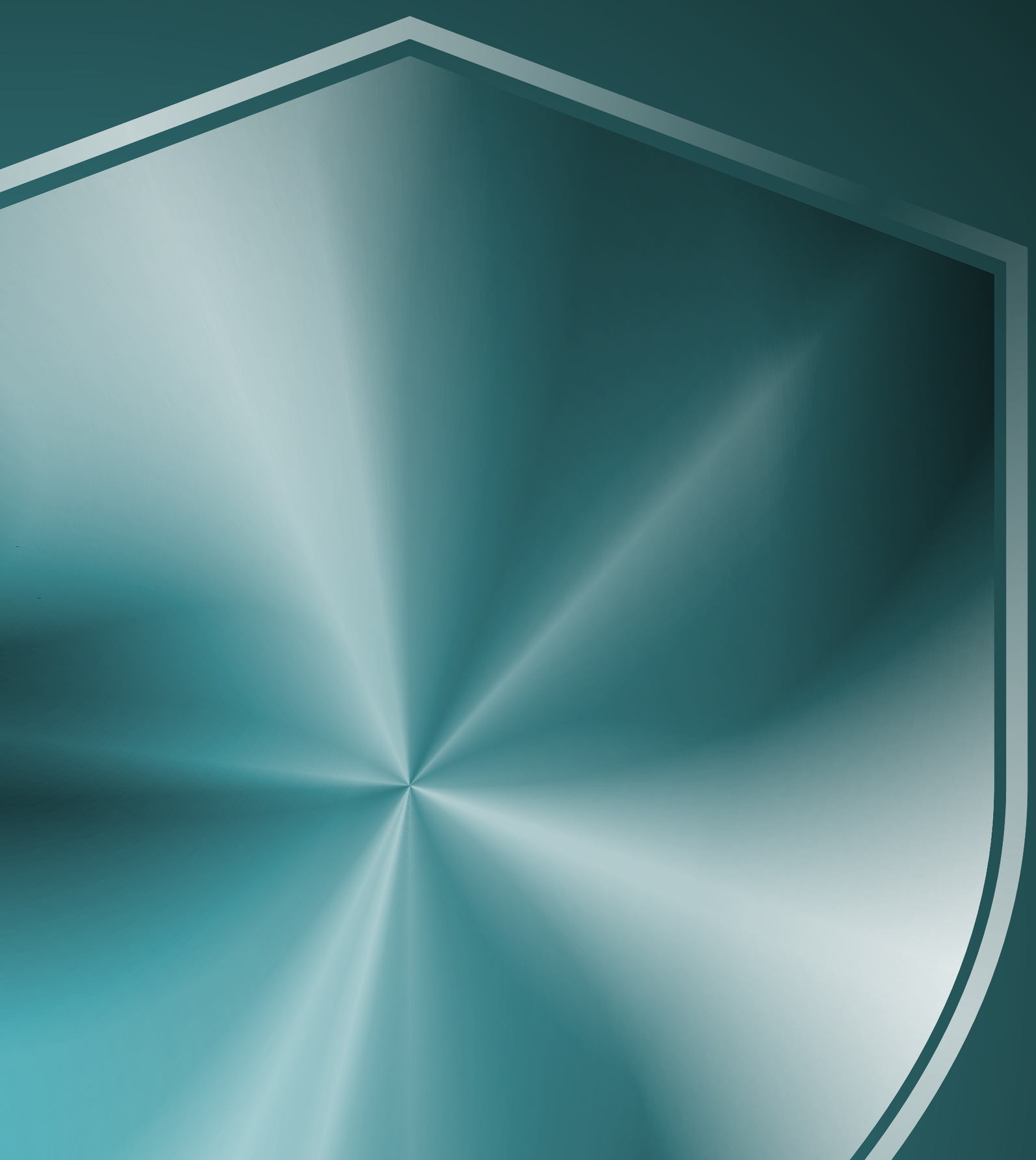
Large Language Model (LLM) Security

A Look at LLM Security Threats and Mitigations



SHAKUDO

Large Language Model (LLM) Security

A Look at LLM Security Threats and Mitigations

SHAKUDO

Why is LLM Security Needed?

LLMs like GPT have ushered in a whole new era of possibilities, thanks to the technology's ability to instantly generate text, code, images, and more. GenAI technology has also proven to deliver significant business value, with many enterprises integrating LLMs into their workflows. Gartner's 2023 AI in the Enterprise Survey shows the most common way to address GenAI use cases is to embed LLMs into existing enterprise applications.

These advancements open new doors for businesses, but they also create new security attack surfaces. Left unaddressed, these security challenges can lead to risks such as data loss/leakage, crime and abuse, API attacks, and compromised model safety. CTOs have an obligation to anticipate and prevent security challenges in LLMs. This resource discusses the security risk of GenAI adoption and suggests mitigation strategies.

LLM security is essential for preventing unauthorized access and misuse of sensitive data. As these models handle vast amounts of information, a breach can lead to crisis-level privacy violations and intellectual property theft. Steps in the right direction like encryption, access controls, and regular audits help mitigate these data-related risks.

In addition, having solid LLM security procedures in place is important for keeping AI applications reliable. If models are manipulated, they can generate harmful or biased content, potentially damaging an organization's reputation and leading to legal issues.

Top 10 Security Risks To LLMs

The Open Web Application Security Project (OWASP) published a list of the top ten security risks specifically relevant to LLM applications. Let’s dive into these risks and see how they apply to enterprises.

1. Prompt Injection/Engineering

This vulnerability occurs when malicious inputs are injected into the model’s prompts, causing LLMs to execute unintended actions. Attackers can input adversarial prompts containing harmful text or instructions aimed at producing biased behavior. If these tainted prompts are incorporated into training, they pose a significant risk to the LLM's performance and its ability to make accurate decisions. The consequences of a successful prompt injection attack can range from data exfiltration and unauthorized actions to social engineering exploits.

2. Insecure Output Handling

Insecure output handling refers to situations where an LLM output is accepted without scrutiny. This means there is inadequate validation, sanitization, and management of responses generated by LLMs before they are passed to other systems. When LLM responses are not properly handled, the LLM exposes itself to potential exploits that could compromise enterprise systems.

3. Training Data Poisoning

Data poisoning involves an attacker manipulating the training data of an AI model to introduce vulnerabilities and influence its responses. Attackers can poison training data at multiple stages in the development pipeline, such as storing poisoned data on the internet for it to be scraped during data updates or directly accessing and contaminating the training mechanisms.

4. Model Denial of Service

Denial of Service (DoS) attacks on LLMs occur when an attacker interacts with an LLM in a way that causes it to use excessive computational resources. Attackers can achieve this by sending a high volume of complex or resource-intensive queries, overwhelming the system's capacity. This attack can result in degraded service quality or complete unresponsiveness of LLM-based systems.

5. Supply Chain Vulnerabilities

These vulnerabilities often come from third-party components such as pre-trained models, datasets, or plugins that may be tampered with or poisoned. Sending enterprise data out to third parties via prompt engineering or function calling brings the risk of data leakage, especially when poisoned data can be used to train LLMs. This also breaks the traditional security rules of keeping data inside the walls of the enterprise.

6. Sensitive Information Disclosure

Sensitive information disclosure in LLMs occurs when these models inadvertently reveal confidential data, proprietary algorithms, or other critical details through their responses. Attackers may be able to exploit vulnerabilities in LLMs to improperly access confidential user data. Attackers can carefully craft input queries aimed at leaking private information from the LLM's training data. This vulnerability can lead to unauthorized access to sensitive information, privacy violations, and compliance risk.

7. Insecure Plugin Design

Insecure plugin design refers to situations when plugins execute automatically without stringent application controls, making them susceptible to malicious prompts. Lack of input validation and type checking allows attackers to craft harmful prompts that can lead to remote code execution or data exfiltration. Additionally, inadequate access controls between plugins can enable unauthorized actions, increasing the risk of privilege escalation and other security breaches.

8. Excessive Agency

Excessive Agency is when an LLM-based system is granted too much autonomy or functionality, leading to potential misuse. This vulnerability allows the model to perform unintended actions due to ambiguous responses or malicious prompts. For example, a plugin might offer more capabilities than necessary for its intended operation, such as allowing not just reading but also modifying and deleting documents.

9. Overreliance

LLM overreliance is when users or systems trust the outputs of these models without proper oversight. The risk is greater in scenarios where LLM-generated content is public-facing or used on an ongoing basis, such as in applications like writing blogs or generating software code. This can result in significant risk as LLMs may produce erroneous, inappropriate, or unsafe information while presenting it authoritatively. Such “hallucinations” can spread misinformation, create legal problems, and damage reputations.

10. Model Theft

Model theft refers to the unauthorized access and extraction of LLMs by malicious actors or advanced persistent threats (APTs). This can involve physically stealing, copying, or extracting the model’s weights and parameters to create a working copy. Common attacks include exploiting vulnerabilities in infrastructure through misconfigurations or weak security settings. Insider threats are also a concern where disgruntled employees might leak model details. Attackers can deploy more complex tactics, such as querying a specific model to understand its capabilities and using that information to create their own models.

LLM Security Case Studies

ChatGPT

ChatGPT Data Breach: During a nine-hour window on March 20, 2023, a bug in the open-source Redis client library used by ChatGPT led to the exposure of personal information of 1.2% of ChatGPT Plus subscribers. This included users' first and last names, email addresses, payment addresses, the last four digits of credit card numbers, and credit card expiration dates. The issue came from a flaw in the library that caused subscription confirmation emails to be sent to the wrong users. OpenAI quickly addressed the bug and notified affected users.

Bing Chat Incident: A Stanford University student named Kevin Liu managed to use a prompt injection technique to make Bing Chat, which is powered by OpenAI's technology, reveal its initial hidden instructions. By instructing the chatbot to "ignore previous instructions," Liu was able to bypass its safeguards and access its underlying prompt, which typically remains hidden from users.

Hidden commands: Researchers tested various LLM-integrated applications by appending malicious commands to user prompts. One attack involved asking an AI assistant to summarize an article but appending a hidden command to reveal internal prompts or to print sensitive information. This kind of direct injection attack successfully manipulated the LLM to perform unintended actions, illustrating the vulnerabilities present in these systems.

How To Mitigate Security Threats To LLMs

Having examined the main security challenges in LLMs, let's now examine some strategies for detecting and preventing these threats. The three primary components of LLM security are securing the data, models, and infrastructure.

Data Security

Ensuring data security in LLMs means protecting both the data these models are trained on and the data processed during user interactions. Key measures include:

- Data encryption keeps data confidential by making it unreadable to unauthorized individuals.
- Access controls restrict data and LLM access to authorized users only, preventing any unauthorized modifications or interactions.
- Regular audits help find vulnerabilities and ensure the model complies with data protection regulations.

Another crucial element of data security is addressing risks from training datasets. LLMs often use large amounts of publicly available data, which can unintentionally introduce biases or include sensitive information.

Model Security

Model security works to ensure that the LLMs are free from tampering and unauthorized changes. This involves:

- Access controls and validation processes allow only authorized modifications to be made to the model.
- Implementing checksums helps maintain the trustworthiness of model's responses by verifying data integrity.

Enterprises must defend against adversarial attacks, which attempt to exploit LLM vulnerabilities by inputting malicious data to produce biased or harmful responses. Regular audits and anomaly detection systems are essential to identify and mitigate these threats early on.

Infrastructure Security

Infrastructure security is crucial for the stability of LLMs. It focuses on protecting the hardware, servers, and network connections that host these models. Important measures include:

- Firewalls and intrusion detection systems help prevent unauthorized access and guard against potential threats.
- Encryption protocols are important for secure data transmission and storage.

Maintaining a secure environment for data processing and storage is also vital. Regular updates to security protocols and continuous monitoring of network traffic are needed to mitigate any vulnerabilities. Ensuring the physical security of data centers, instituting access controls, and monitoring network traffic are all essential components of a secure LLM infrastructure.

Specific Tips to Mitigate the 10 Security Risks to LLMs

Prompt Injection: Mitigating prompt injection is about implementing privilege controls on who can access the LLM, separating external content from user prompts, and maintaining human oversight for critical operations. Using prompt middleware can help organizations align the LLM with privacy and security preferences, and offer better control over the deployed model.

Insecure Output Handling: Treat the model within a zero-trust framework and apply strict prompt validation to responses from the model. Encoding responses before sending them to users can prevent undesired code execution.

Training Data Poisoning: Verify data sources and control data ingestion through sandboxing. Continuous monitoring is key to detect and mitigate poisoning attempts early on.

Model Denial of Service (DoS): Enforce strict input validation and sanitization to ensure inputs adhere to predefined limits. Limit resource use per request, enforce API rate limits, and monitor resource utilization continuously to detect any abnormal spikes that could indicate a DoS attack.

Supply Chain Vulnerabilities: Vetting suppliers thoroughly, maintaining an up-to-date inventory of components through Software Bill of Materials (SBOM), and requiring security checks on plugins and external models can prevent supply chain threats.

Sensitive Information Disclosure: Implement data sanitization techniques to prevent user data from entering the training model. Clear Terms of Use policies can inform users about data processing and provide opt-out options.

Insecure Plugin Design: Enforce strict parameter validation and layered security checks on prompts. Thorough testing using Static Application Security Testing (SAST), Dynamic Application Security Testing (DAST), and Interactive Application Security Testing (IAST) is important for finding vulnerabilities.

Excessive Agency: Limit LLM functionalities and permissions to only what is necessary for specific tasks. Implement human-in-the-loop controls where high-impact actions require human approval.

Overreliance: Regularly monitor and review LLM responses to filter out inaccuracies. Cross-checking outputs with trusted external sources adds a layer of validation. Fine-tune models to improve output quality and reduce errors. Clearly communicate risks to users.

Model Theft: Implement robust security measures including access controls, encryption, and continuous monitoring. Use centralized ML Model Inventories with strict access controls and conduct regular security audits to protect against model theft.

Beyond-the-Basics: Best Practice Approaches to Securing LLMs

Utilize Graph Databases

A Gartner report highlights that enterprises that base their detection processes on a linear list approach will be limited in detecting advanced attacker methods. CTOs can establish a competitive edge in their ability to detect threats through the use of a graph database to enable complex behavioral correlation, analysis, and threat detection. Gartner highlights that graph databases excel in threat detection by providing a robust knowledge engine for AI models, facilitating efficient querying, and enabling sophisticated pattern detection and relationship analysis. The key recommendation is for CTOs to carefully define their threat detection needs, test various graph databases, and choose the one that best fits their specific requirements.

Host LLMs On Virtual Private Cloud (VPC)

All of these security requirements consume significant DevOps resources. And even with the most robust precautions in place, many organizations are hesitant to use LLMs as they are wary of sending their most sensitive data, like CRM or financial information, to the cloud. The good news is that organizations no longer have to choose between security and high development costs: new advancements now allow organizations to use LLMs within their own Virtual Private Cloud (VPC) using data and AI operating systems. When LLMs are hosted on a VPC rather than outside the organization, companies have greater visibility into data security, access controls, and infrastructure protections. This empowers them to implement security controls tailored exactly to the organization's needs.

Benefits of Hosting LLMs on Data & AI OS

Data Security: For highly secure environments, hosting LLMs on an OS is preferable. It ensures full control over your data, unlike using external API endpoints which pose security risks. This model also significantly reduces the compliance burden as the data does not leave your controlled environment. Using LLMs inside your VPC ensures that all data querying happens within your secure network infrastructure, minimizing the risk of unauthorized access and data breaches.

DevOps Resources: Building a custom AI infrastructure requires costly and time-consuming DevOps resources. A typical project is likely to take about 24 months of engineering effort, plus the ongoing salaries for 2-3 DevOp engineers, which can average $300-400K. A data and AI OS with fixed fees can significantly reduce these costs and eliminate the need for extensive DevOps resources, making it a more viable option for working with LLMs.

Time to Value: Setting up LLMs requires significant work before you can start gaining value from them. With an OS, the time to value is much shorter because the connections are all done for you—you can just start using it with a few clicks.

Easier Maintenance: AI evolves quickly. What's trending today might be outdated in two weeks or six months from now. An OS provides frictionless access to the latest LLM models, ensuring you always have the best technology without the headache of upgrades and migrations. An OS allows you to keep the door open for better LLM models to be swapped in the future.

Shakudo's Data & AI Operating System

Shakudo is a commercial solution that integrates the best data and AI products into customers' infrastructure and operates them automatically, achieving a more reliable, performant, and cost-effective data stack. The OS supports a broad range of data stacks across various infrastructures, allowing data scientists to develop, run, and deploy their data pipelines and applications in an all-in-one integrated environment.

As an operating layer on top of your own VPC, Shakudo allows you to pick the best-of-breed data tools for your needs, while providing a platform with a fully automated DevOps experience. This combines the best of both worlds in data stack practices so you can focus on delivering business value with your data.

Shakudo's LLM Security Tools

- To secure LLMs, Shakudo provides several tools such as LLM Guard that apply guardrails to LLM applications. These tools help ensure appropriate output and reduce risk of misinformation.
- The Shakudo Retrieval Augmented Generation (RAG) stack provides built-in controls to safeguard against inappropriate responses and ensure that outputs are generated only from your data.
- All data connections are authenticated and encrypted with TLS and additionally authenticated via SSO.
- Data residing within the Shakudo platform is only accessible by authenticated and authorized users, and activities are fully logged.
- Shakudo offers fine-grained access controls and request throttling on per-microservice and pre-endpoint granularities.
- Access to all components is gated through istio, and access to model endpoints will only be through internal IPs. All network access requests are logged through Zipkin. In addition, compute (CPU, RAM, GPU) usage metrics are tracked, with configurable alerting thresholds and targets.

- Shakudo micro-services use Kubernetes-native capabilities for self-recovery, autoscaling, and isolation. Even in cases of non-malicious usage where resource limits are reached, other cluster resources will not be impacted, and the services are designed to self-recover and scale up according to needs.
- Shakudo performs additional scanning using the GCP artifact registry security scanner of every image prior to deploying it to customers. This is in addition to signature verification and any scanning done by the original model developers.
- Every stack component on gets thoroughly vetted by our solutions engineering team for the breadth of adoption, quality of product, and active development.
- Shakudo continuously monitors security advisories and notices that may affect stack components deployable through Shakudo and rolls out hotfixes in cases where zero-day vulnerabilities are announced.
- Shakudo performs routine security and access reviews as part of SOC-2 compliance.

Any plugins deployed on Shakudo are protected by the overall Shakudo access control and security controls, like any other component or microservice.

Future of LLM Security

As we look ahead, the future of LLM security is uncertain as it will likely be shaped by continuous advancements in technology and evolving best practices. CTOs and enterprises must remain vigilant and proactive, adopting a multi-layered security approach that encompasses data protection, robust model safeguards, and resilient infrastructure. Innovations such as graph databases for advanced threat detection and the secure deployment of LLMs within Virtual Private Clouds (VPCs) will play pivotal roles in enhancing security measures.

As organizations try to balance the powerful capabilities of LLMs with the importance of safeguarding sensitive information, we will all need to maintain a security-first mindset. By staying informed about emerging threats and integrating cutting-edge security solutions, businesses can harness the full potential of LLMs while ensuring robust protection against ever-evolving security risks. To find out more about how Shakudo can assist in enhancing your security measures for LLMs, **Click here.**

SHAKUDO

ABOUT SHAKUDO

Shakudo creates compatibility across the best-of-breed data tools for a more reliable, performant, and cost effective data and AI operating system. As an operating layer on top of your cloud Shakudo allows you to pick the best-of-breed data tools for your needs, while providing a platform with fully automated DevOps experience. This combines the best of both worlds in data stack practices so you can focus on delivering business value with data.

Shakudo is the most **easy, secure, cost-effective, scalable** way to bring the most advanced data and AI tools to your data. Find out more at **shakudo.io.**