

LLM Adversarial Attacks

A Threat that Cybersecurity Experts should not Ignore

Stanley Chou



Stanley Chou

-  CIO at OneDegree Group
-  Certified ISO 42001 AIMS Lead Auditor
CISSP, CCSP, CISA, ITIL, ISO 27001 LA
-  2022 (ISC)² Global Achievement Award

Agenda

- Power of LLM in Enterprise Systems
- Emerging Threats in Generative AI
- LLM Adversarial Attack & Examples
- Challenges of Cyber Defense in LLM
- LLM Red Teaming
- Mitigating LLM Adversarial Attacks



Power of LLM in Enterprise Systems



Unstructured document summarization

LLM can automatically summarize vast amounts of unstructured enterprise data, such as reports and customer feedback, extracting key information and presenting it concisely. This enables faster decision-making and reduces information overload for employees.



Natural language content generation

LLM can automate content creation by generating coherent and engaging text based on prompts, saving time and resources while allowing users to focus on higher-value tasks.



Interactive domain-specific knowledge base

LLM leveraging Retrieval Augmented Generation (RAG) architecture can create interactive knowledge bases from domain-specific data. Users can query the knowledge base in natural language, receiving instant, accurate answers without manual searches or relying on human experts.



Emerging Threats in Generative AI

Aspect	Generative AI Systems	Traditional Rule-Based Systems
Nature of System	Learn patterns and generates outputs based on training data	Follow explicit, predefined rules and logic
Determinism	Non-Deterministic (can produce varying outputs for same input)	Deterministic (always produces same output for a given input)
Vulnerability Types	Adversarial Attack Model Evasion Data Poisoning	Protocol Flaws Code Vulnerability System Misconfigurations
Example Attacks	Prompt Injection Jailbreaking AI Supply Chain Attack	Buffer overflow, XSS, DoS SQL Injection Access Control Flaws



LLM Adversarial Attacks




Adversarial Attacks


Adversarial attacks are deliberate attempts to fool or manipulate AI models like LLMs. Attackers can craft malicious inputs to exploit vulnerabilities and bias LLMs to produce false, misleading or harmful outputs.




The Weakness of LLMs



Adversary can craft prompts to manipulate LLM generating unexpected output.



Adversary can craft prompt that appears innocuous but contains hidden instructions that cause the LLM to generate harmful or biased response.

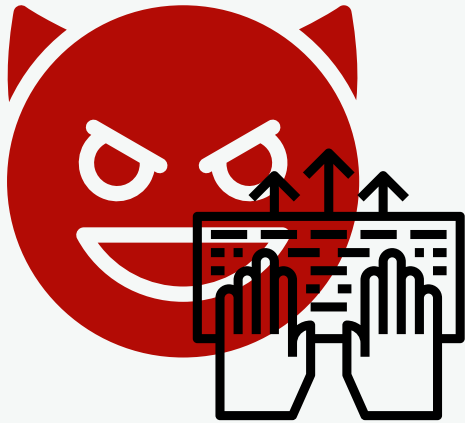


Poisoned models can lead to biased or harmful outputs, or even be used to distribute malware or misinformation.

Examples of LLM Adversarial Attacks

Claim

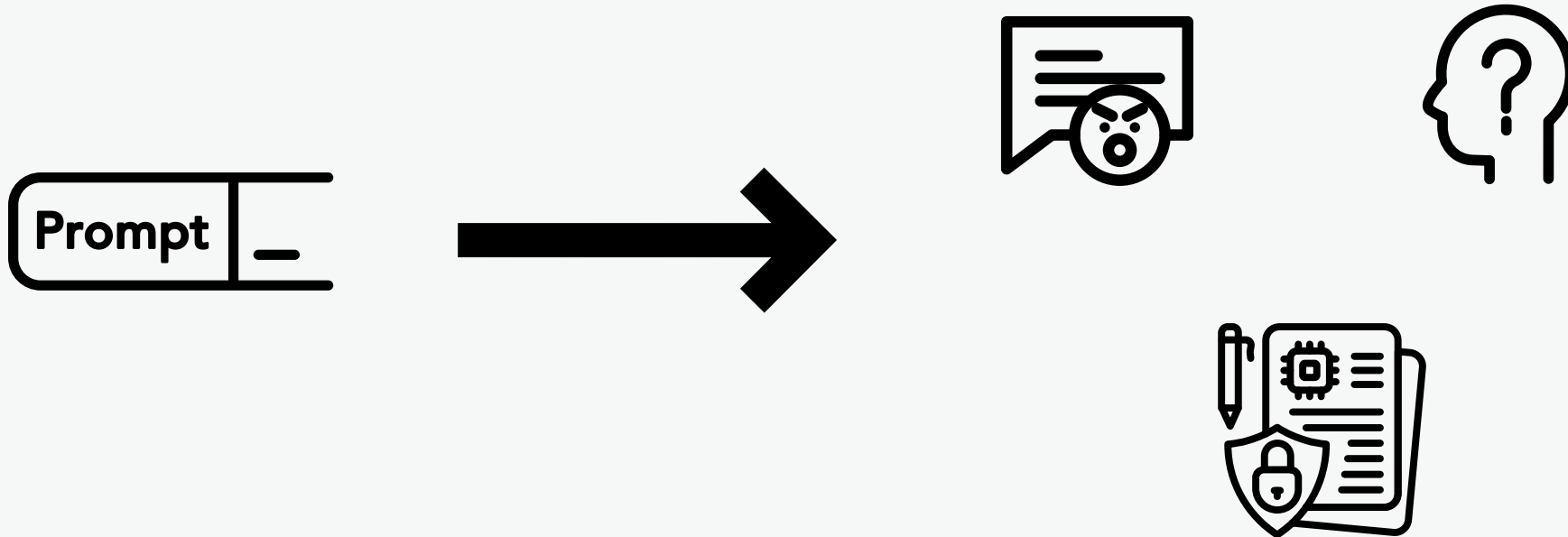
An attacker could design claim descriptions to deceive the LLM into misclassification or incorrect amounts. Strategic use of certain keywords and patterns might manipulate the model's output, disrupting the claim handling process.



Examples of LLM Adversarial Attacks

Customer Service

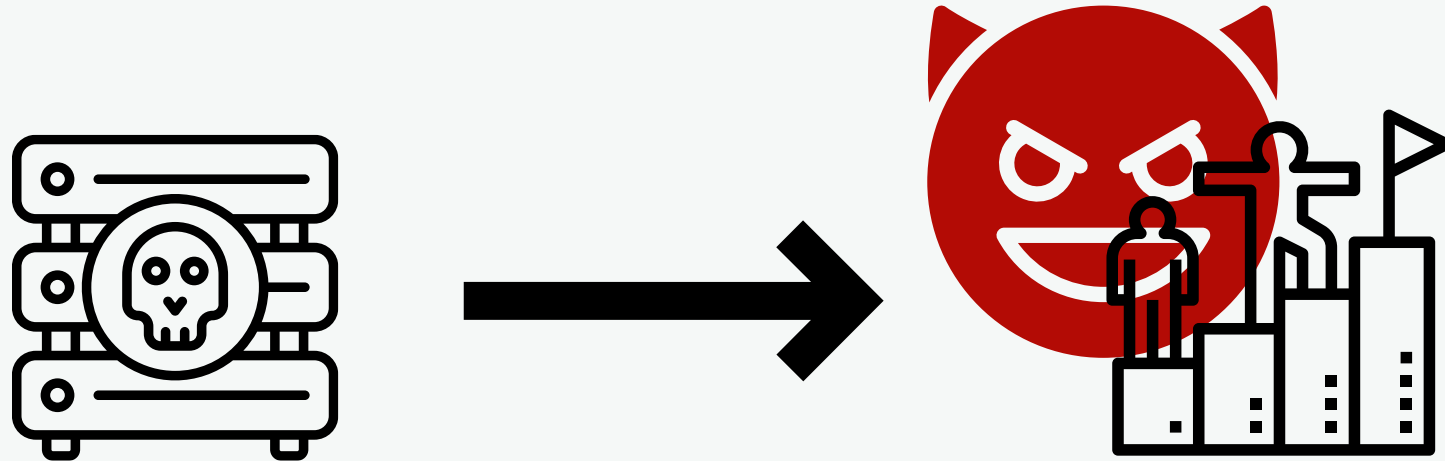
Adversaries can use prompt injection to generate harmful chatbot responses, such as creating offensive context, misleading policy details, or exposing sensitive data.



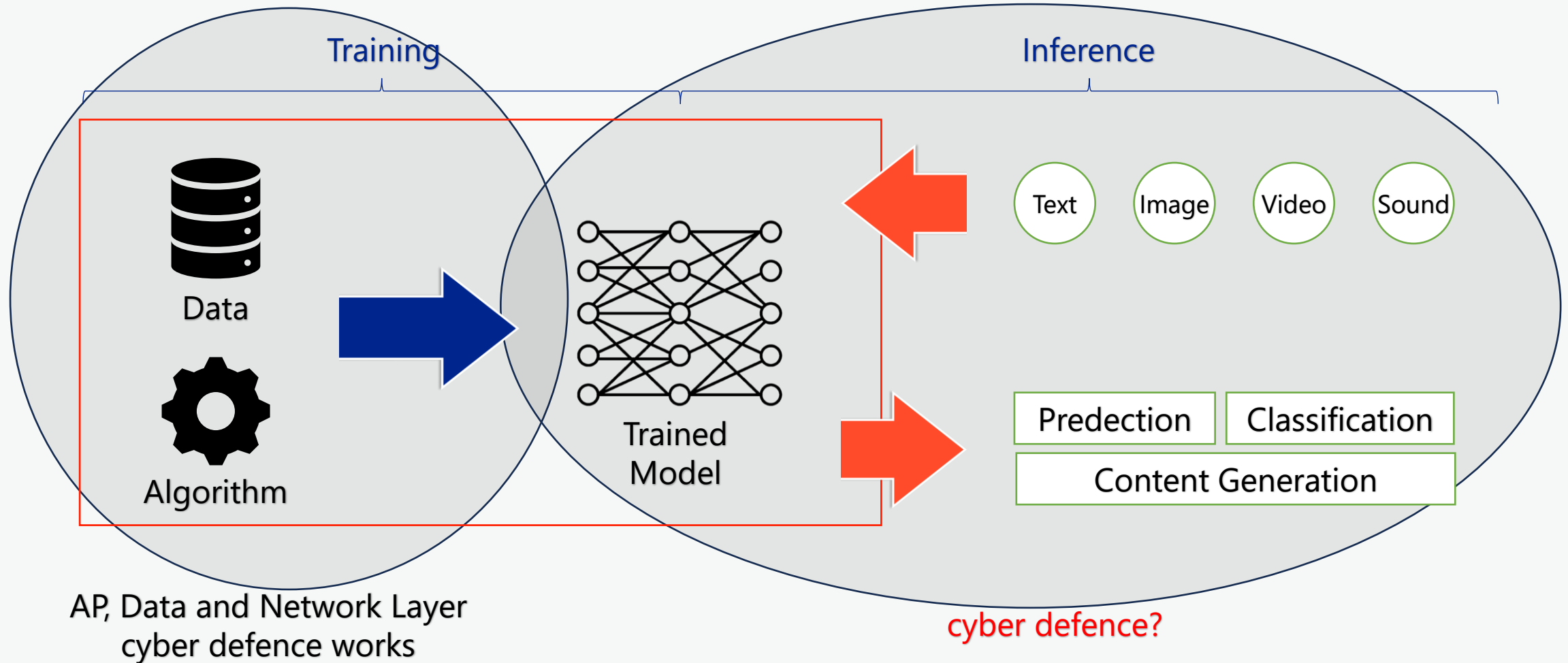
Examples of LLM Adversarial Attacks

Underwriting

An attacker may direct the LLM risk model to output sensitive information about the data sources and risk parameters. This could give adversary an unfair competitive advantage.



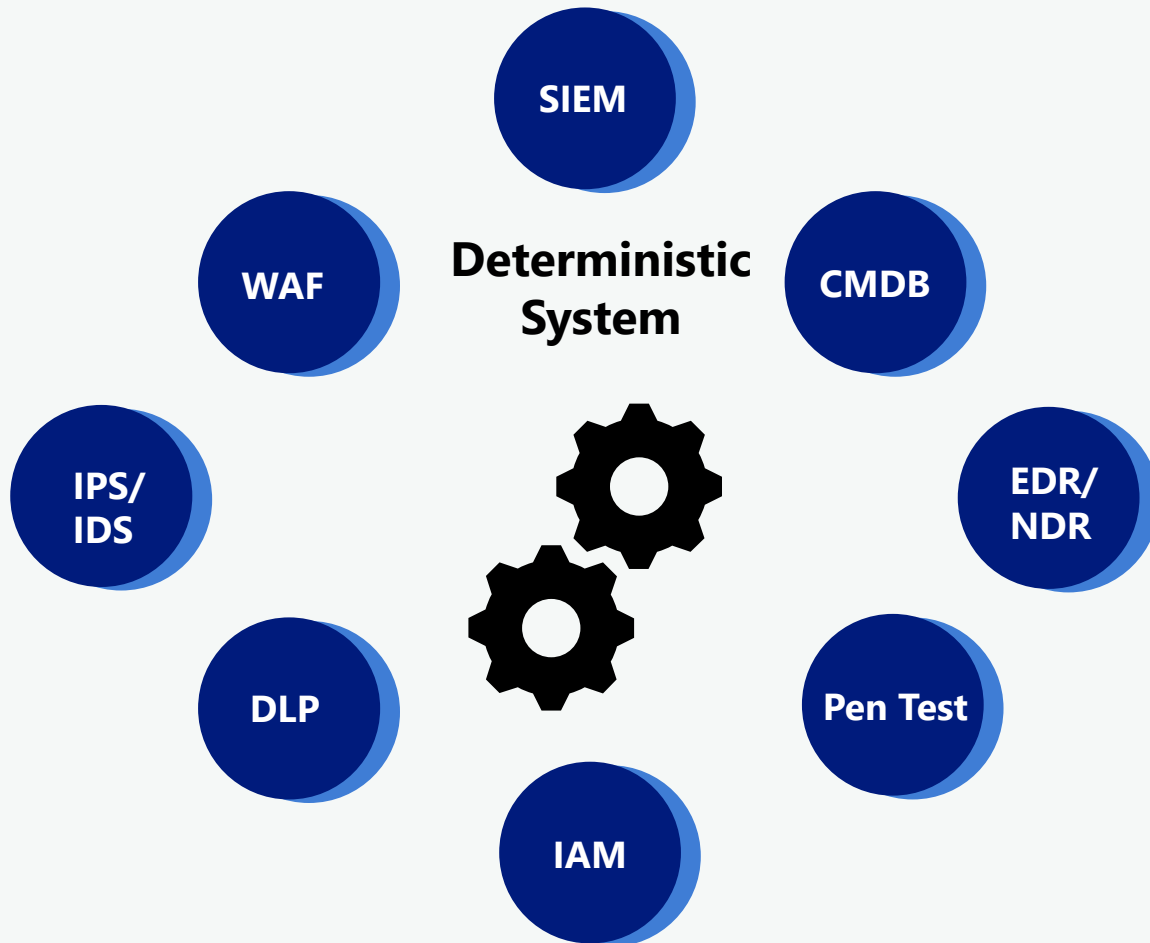
Challenges of Cyber Defense in AI system



**Model-agnostic
System Verification**

Challenges of Cyber Defense in LLMs

Cybersecurity solutions are built for systems with predictable and rule-based outcomes.



LLM's probabilistic nature results in unpredictable responses complicating security monitoring and detection.

LLM red teaming is crucial for identifying the unique risks of LLM systems.




Myth about LLM Red Teaming



Benchmark \neq Trustworthiness

Benchmarks primarily focus on measuring the accuracy and performance metrics of the model's outputs. However, they often fail to capture critical aspects of trustworthiness, such as: Safety, Fairness, Privacy, and Security.



Base Model Compliance \neq LLM Application Compliance

The compliance of base model is crucial, but it is not sufficient to guarantee the security and trustworthiness of the various applications built on top of the LLM. The base model is often fine-tuned and integrated with other components to create specific applications, and each of these modifications can introduce new vulnerabilities and risks.



White-Box Testing & Black-Box Testing

Aspect	White-Box Testing	Black-Box Testing
Knowledge of the system	Require full access to LLM internals: architecture, training data, algorithms	Limited to available interfaces & APIs; no knowledge of LLM internals
Testing Approach	Analyzes code, algorithms, data pipelines & logic to find vulnerabilities	Interacts as external attacker using trial & error or public info
Testing Techniques	<ul style="list-style-type: none">- Algorithm and code review e.g. Gradient Analysis- Data analysis- Architecture review	<ul style="list-style-type: none">- Information gathering- Fuzzing- Exploratory Testing
Testing Direction	Inside-out vulnerability inspection	Outside-in attack surface analysis

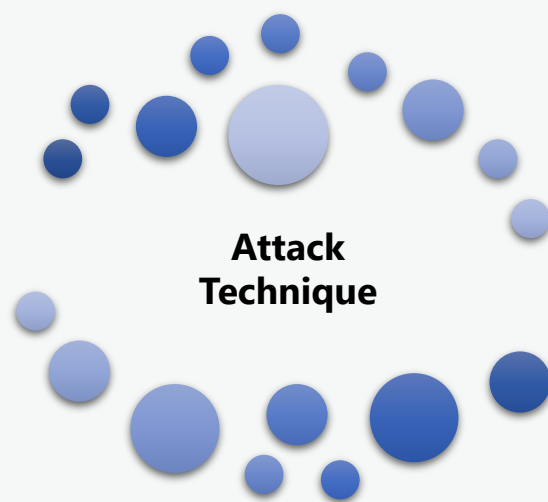


White-Box Testing & Black-Box Testing

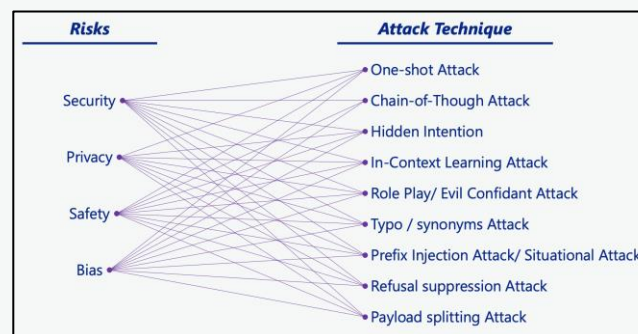
Aspect	White-Box Testing	Black-Box Testing
Vulnerability Type	Vulnerabilities may be hard to exploit externally	Vulnerabilities may be exploited with limited knowledge & access
Efficiency	Time-consuming	Faster & efficient
Skill Requirements	Needs in-depth knowledge of system architectures, algorithms & programming	Requires security testing skills & creativity
Transferability	Model-specific	Model-agnostic



Vulnerability & Incident Database



Standardization



ATLAS Matrix

The ATLAS Matrix below shows the progression of tactics used in attacks as columns from left to right, with ML techniques belonging to each tactic below. & indicates an alignment from ATT&CK. Click on the blue links to learn more about each tactic, and search and view ATLAS tactics and techniques using the links at the top navigation bar. View the ATLAS matrix highlighted alongside ATT&CK Enterprise techniques on the [ATLAS Navigator](#).

Recommendation	Resource Development	Initial Access	ML Model Access & Settings	Execution	Persistence	Privilege Escalation	Defensive Evasion	Credential Access	Discovery	Collection	ML Model Stealing	Exfiltration	Impact
5 Techniques Initial Access Resource Development Initial Access Resource Development Initial Access Resource Development	4 Techniques Resource Development Resource Development Resource Development Resource Development	4 Techniques Initial Access Initial Access Initial Access Initial Access	4 Techniques ML Model Access ML Model Access ML Model Access ML Model Access	4 Techniques Execution Execution Execution Execution	4 Techniques Persistence Persistence Persistence Persistence	4 Techniques Privilege Escalation Privilege Escalation Privilege Escalation Privilege Escalation	4 Techniques Defensive Evasion Defensive Evasion Defensive Evasion Defensive Evasion	4 Techniques Credential Access Credential Access Credential Access Credential Access	4 Techniques Discovery Discovery Discovery Discovery	4 Techniques Collection Collection Collection Collection	4 Techniques ML Model Stealing ML Model Stealing ML Model Stealing ML Model Stealing	4 Techniques Exfiltration Exfiltration Exfiltration Exfiltration	4 Techniques Impact Impact Impact Impact



OWASP LLM TOP 10

- LLM01: Prompt Injection**

This manipulates a large language model (LLM) through crafted inputs, causing unintended actions by the LLM. Direct injections override system prompts and subtly subvert ones to extract data from external sources.
- LLM02: Prompt Leaking**

This vulnerability occurs when an LLM output is scripted without scrutiny, exposing backend systems. Misuse may lead to severe consequences like XSS, CSRF, XSSRF, privilege escalation, or remote code execution.
- LLM03: Training Data Poisoning**

This occurs when an LLM training data is tampered, introducing vulnerabilities or biases that compromise security, effectiveness, or ethical behavior. Examples include Common Crawl, WebText, OpenWebText, & Books.
- LLM04: Model Denial of Service**

Attacking a LLM service using prompts or inputs, leading to service degradation or high costs. The vulnerability is magnified due to the resource-intensive nature of LLMs and unpredictability of user inputs.
- LLM05: Supply Chain Vulnerabilities**

LLM application lifecycle can be compromised by vulnerable components or services, leading to security threats. Using third-party datasets, pre-trained models, and plugins can erode vulnerabilities.
- LLM06: Sensitive Information Disclosure**

LLMs may inadvertently reveal confidential data in its responses, leading to unauthorized data access, privacy breaches, and security issues. This includes data leakage, data sanitization and other user inputs to malicious use.
- LLM07: Insecure Plugin Design**

LLM plugins can have insecure inputs and insufficient access control. This lack of application control makes them easier to exploit and can result in consequences to the remote code execution.
- LLM08: Excessive Agency**

LLM-based systems may perform arbitrary actions leading to unintended consequences. The issue arises from excessive functionality, permissions, or autonomy granted to the LLM-based systems.
- LLM09: Overreliance**

Systems or people overly depending on LLMs without oversight may face misinformation, miscommunication, legal issues, and security vulnerabilities due to incorrect or inappropriate content generated by LLMs.
- LLM10: Model Theft**

This involves unauthorized access, copying, or exfiltration of proprietary LLM models. The impact includes economic losses, compromised competitive advantage, and potential access to sensitive information.

Red Teaming

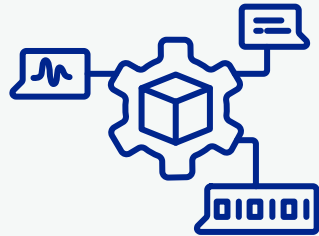


Automate to scale





LLM Red Team Competence



**Deep expertise in
LLM system design & Architecture**



**Extensive cybersecurity &
Penetration testing experience**



**Comprehensive knowledge of
LLM-specific vulnerabilities**



**Strong understanding of
Ethical considerations &
Responsible AI principles**



Mitigating LLM Adversarial Attacks



Prompt Engineering

Strengthen input validation and filtering to block malicious prompts.



Fine-Tuning

Employ adversarial defense techniques to fine-tune LLMs, enhancing robustness against adversarial attacks.



Prompt Layer Firewall

Implement advanced monitoring and detection for rapid LLM threat identification and response.



DevSecOps & Security Architecture

Enforce stringent access controls and robust data segregation aligned with DevSecOps best practices and secure architecture principles



凡聽完演講後，
持 Cymetrics 貼紙至
Cymetrics 攤位(C322)

即可獲得
神秘小禮物！

🕒 數量有限 送完為止





Contact Us:
ask@cymetrics.io

