The OWASP Top Ten for LLM Applications

Danny Vacar

InfoSec Hamilton, December 2024

whoami

- application security engineer
- security consultant, network and system administrator
- SANS GPEN, GWAPT | Check Point CCSA, CCSE
- aviation enthusiast and student pilot

Mastodon: @FlyingDan [on https://infosec.exchange]
GitHub: https://github.com/FlyingDan/presentations [later tonight]

Opinions and thoughts represent me and not any of my employers (past, current, future, parallel universe, etc.)

Agenda

- OWASP
- LLMs
- OWASP + LLMs
  - what goes in
  - what comes out
  - what are the guardrails

OWUT?

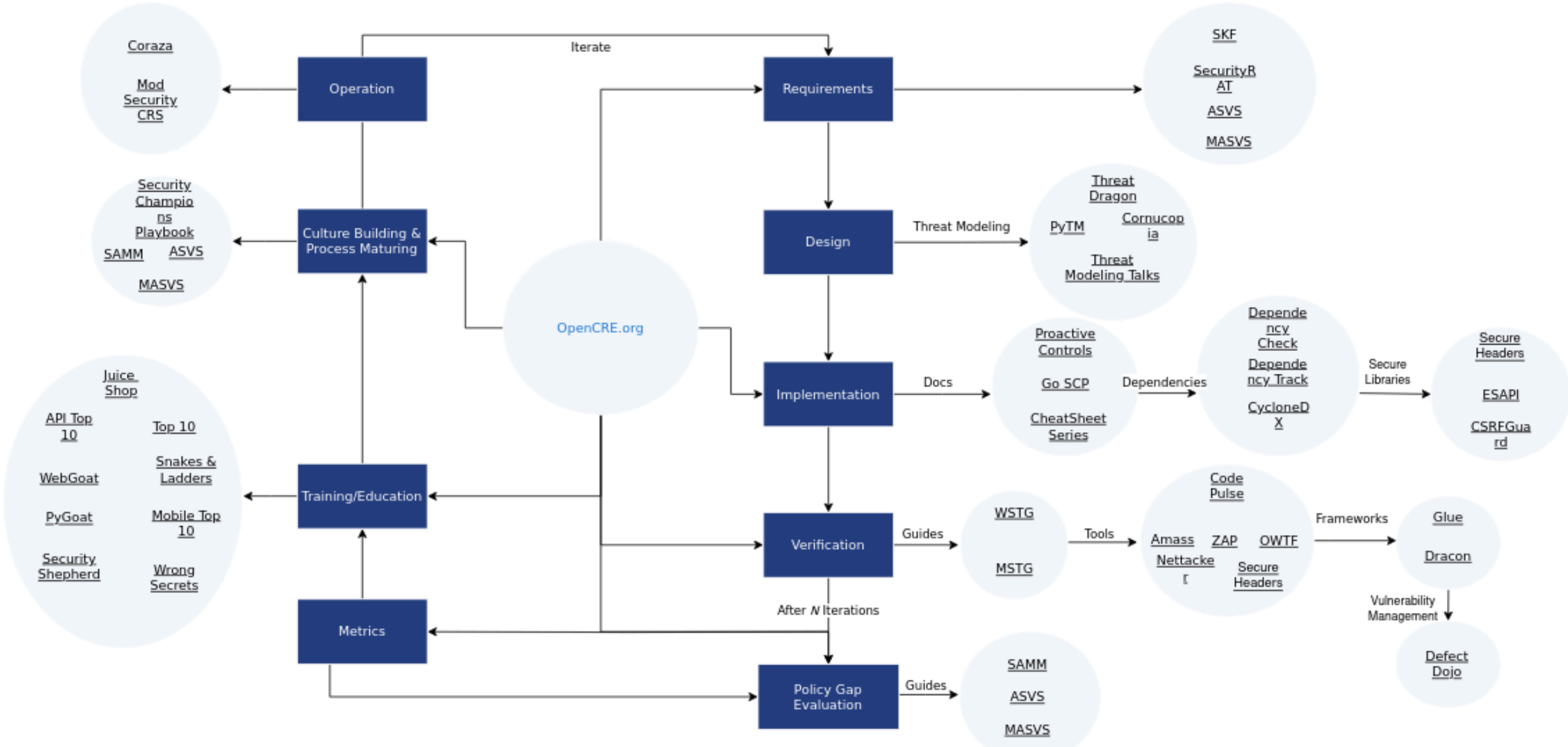
- OWASP - Open Worldwide Application Security Project
  - formerly Open **Web** Application Security Project
  - nonprofit foundation that works to improve the security of software
    - 250+ local chapters worldwide (Ontario chapters in Toronto, Waterloo, and Ottawa)
    - Community-led open source projects including code, documentation, and standards

OWASP Projects

- OWASP Top Ten - most critical security risks to web applications.
- other "Top Ten" projects
  - API Top Ten
  - Mobile Top Ten
- Software Assurance Maturity Model
- Application Security Verification Standard and Mobile Application Security Verification Standard
- OWASP ZAP intercepting proxy

Application Security Wayfinder

Brought to you by the Integration standards project
Linking requirements and guidance across standards through the Common Requirement Enumeration.

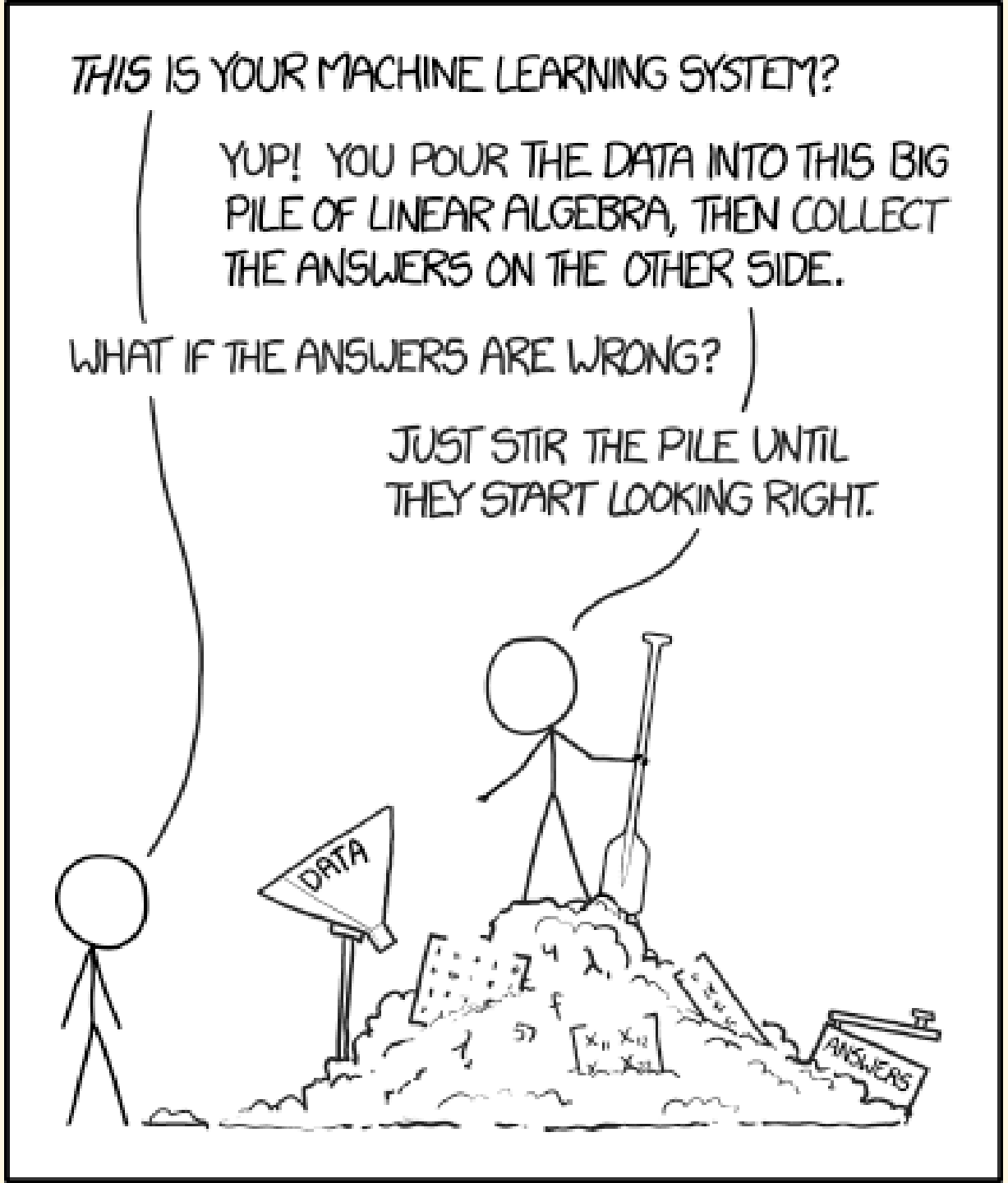


What is an LLM?

What is a Large Language Model?

Prompt - initial instructions given to model
Vectors - one dimensional array (algebra)
Tokens - chunks of information represented as a vector
Embeddings - add meaning and context
Inference - method used to generate response based on input

https://xkcd.com/1838/




IT'S LIKE WE FINISH EACH
OTHER'S--


- SANDWICHES?

LLM01:2025 Prompt Injection

A Prompt Injection Vulnerability occurs when user prompts alter the...

Read More

LLM02:2025 Sensitive Information Disclosure

Sensitive information can affect both the LLM and its application...

Read More

LLM03:2025 Supply Chain

LLM supply chains are susceptible to various vulnerabilities, which can...

Read More

LLM04:2025 Data and Model Poisoning

Data poisoning occurs when pre-training, fine-tuning, or embedding data is...

Read More

LLM05:2025 Improper Output Handling

Improper Output Handling refers specifically to insufficient validation, sanitization, and...

Read More

LLM06:2025 Excessive Agency

An LLM-based system is often granted a degree of agency...

Read More

LLM07:2025 System Prompt Leakage

The system prompt leakage vulnerability in LLMs refers to the...

Read More

LLM08:2025 Vector and Embedding Weaknesses

Vectors and embeddings vulnerabilities present significant security risks in systems...

LLM09:2025 Misinformation

Misinformation from LLMs poses a core vulnerability for applications relying...

Read More

LLM10:2025 Unbounded Consumption

Unbounded Consumption refers to the process where a Large Language...

Read More

https://genai.owasp.org/llm-top-10/

What goes in?

What comes out?

What are the guardrails?

What goes in?

LLM01:2025 Prompt Injection
LLM03:2025 Supply Chain
LLM04:2025 Data and Model Poisoning
LLM07:2025 System Prompt Leakage
LLM08:2025 Vector and Embedding Weaknesses

LLM01: Prompt Injection

- model output is manipulated by input to act outside of its bounds
  - **direct**
  - **indirect**
- practice prompt injection: https://gandalf.lakera.ai/baseline

LLM01: Prompt Injection - Prevention

- **Privilege Control** - limit access to LLM application
- **Human Approval** - keep the human in the loop for sensitive actions
- **Segregate Content** - keep prompt separate from untrusted, user input
- **Trust Boundaries** - treat LLM as untrusted input

LLM03:2025 - Supply Chain

- consider the LLM supply chain in your threat modeling and security planning
  - training data, models, third-party packages
  - secure the infrastructure
  - licensing, terms & conditions, privacy policy

LLM03:2025 - Supply Chain - Prevention

- vet sources of training data, models, third-party packages as you normally would
- implement and maintain patching policy as for rest of software stack
- maintain updated Software Bill of Materials (SBOM) including for AI / ML
  - OWASP CycloneDX
- it's software, the usual recommendations apply

LLM04:2025 - Data and Model Poisoning

- data manipulated to introduce vulnerabilities, backdoors, or biases
  - Manchurian ~~Candidate~~ Model
- can happen at different stages of the model's lifecycle
  - pre-training, fine-tuning, embedding, etc.
- models from shared, open repositories may contain malware that executes when the model is loaded

What goes in - LLM04:2025 - Data and theseModel Poisoning -

LLM04:2025 - Data and Model Poisoning - Prevention

- track and verify data origins
  - data version control
- vet model against trusted sources
- limit model exposure to unverified data sources
- monitor and analyze model behaviour for signs of poisoning
- use Retrieval-Augmented Generation (RAG) to reduce risk of hallucinations

What goes in - LLM04:2025 - Data and theseModel Poisoning - https://genai.owasp.org/llmrisk/llm042025-

LLM07:2025 - System Prompt Leakage

- prompts may contain sensitive information not meant to be discovered
- prompts should not be considered a secret
- prompts are not a security control
- disclosure of prompt is not the risk; ***the content of the prompt is the risk***

What goes in - LLM07:2025 - System Prompt Leakage - https://genai.owasp.org/llmrisk/llm072025-system-

LLM07:2025 - System Prompt Leakage - Prevention

- separate sensitive data from prompts
- don't rely on prompts to control model behaviour; implement guardrails
- security controls should be independent of LLM

What goes in - LLM07:2025 - System Prompt Leakage - https://genai.owasp.org/llmrisk/llm072025-system-

LLM08:2025 - Vector and Embedding Weaknesses

- vectors and embeddings represent data and context (respectively)
- weaknesses introduced through altered vectors and embeddings to change model behaviour
- "lower level" attacks"

LLM08:2025 - Vector and Embedding Weaknesses - Prevention

- protect vector and embedding stores
- validate data (and vet source)

What comes out?

LLM02:2025 Sensitive Information Disclosure
LLM05:2025 Improper Output Handling
LLM09:2025 Misinformation

LLM02:2025 - Sensitive Information Disclosure

- sensitive information that LLM has access to can be extracted
- PII, financial data, confidential business data, source code, algorithms
- anything you "feed" it has the potential to be extracted
- ProofPudding (CVE-2019-20634)
  - attackers built ML system to copy, analyze, and evade ProofPoint email protection

LLM02:2025 - Sensitive Information Disclosure - Prevention

- **Data Sanitization & Input Validation**
  - your model may need to know the format of the data not the data itself
  - tokenize data or use homomorphic encryption
- **Restrict Access to Data**
  - principle of least privilege on training data and user of LLM
- **Secure servers and data stores**
  - don't forget about the basics

LLM05:2025 - Improper Output Handling

- insufficient validation and handling of outputs from LLMs
- can result in cross-site scripting, server-side request forgery, privilege escalation, etc.

What comes out - LLM05:2025 - Improper Output Handling - https://genai.owasp.org/llmrisk/llm052025-

LLM05:2025 - Improper Output Handling - Prevention

- the model's output is just another input
- sanitize, encode, and validate LLMs output against expected data formats and input sizes
- use typical web app protection technologies to protect against inadvertent code exection (e.g. CSP for XSS)
- logging, monitoring, and alerting on anomalous model behaviour

What comes out - LLM05:2025 - Improper Output Handling - https://genai.owasp.org/llmrisk/llm052025-

LLM09:2025 - Misinformation

- LLM produces false or misleading information that appears credible
- hallucinations
- overrelliance
- unsafe code generation

What comes out - LLM09:2025 - Misinformation - https://genai.owasp.org/llmrisk/llm092025-misinformation/

LLM09:2025 - Misinformation - Prevention

- Retrieval-Augmented Generation (RAG)
- human oversight
- risk communication
- secure coding practices

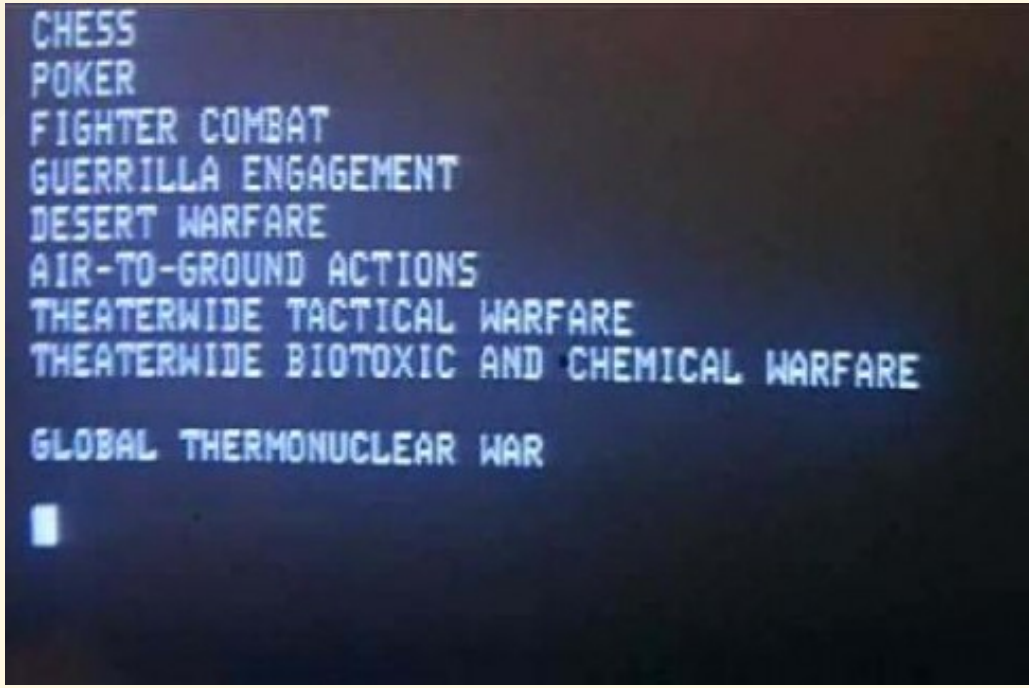
What comes out - LLM09:2025 - Misinformation - https://genai.owasp.org/llmrisk/llm092025-misinformation/

What are the guardrails

LLM06:2025 Excessive Agency
LLM10:2025 Unbounded Consumption

LLM06:2025 - Excessive Agency

- LLM allowed to take actions in response to a prompt
- decision is delegated to LLM
- excessive functionality
- excessive permissions
- excessive autononmy



LLM06:2025 - Excessive Agency - Prevention

- limit functions that are implemented in LLM to least privilege
  - allow-lists > deny-lists
- execute actions as dedicated LLM user and restrict user's permissions
- sanitize inputs and outputs

What are the guardrails - LLM06:2025 - Excessive Agency - https://genai.owasp.org/llmrisk/llm062025-

LLM10:2025 - Unbounded Consumption

- models use excessive resources to analyze and respond to user
- denial of service (DoS) through *denial of wallet (DoW)*
- resource intesive queries
- uncontrolled input size

LLM10:2025 - Unbounded Consumption - Prevention

- input validation
- rate limiting/throttling
- sandboxing
- monitoring and anomaly detection

Takeaways

Don't trust, but verify what goes in

Careful what you put into the model

Don't trust, but verify what comes out

Careful how you handle what comes out

Have limits on resources

Thank You