CYB-4203/6203

Secure and Trustworthy AI

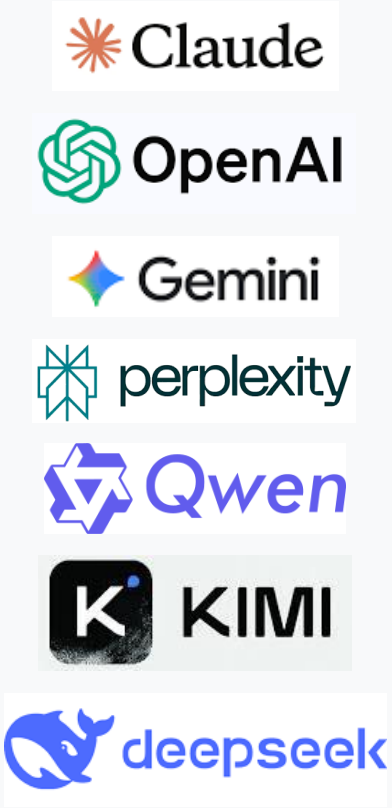
Presentation 11: LLM-specific Vulnerabilities and AI/ML Threat Modeling Frameworks

Wednesday, March 4, 2026

Topics: 6.3, 6.4

Why LLMs are different

- Large Language Models (LLMs) are one of the most impactful, promising, and dangerous technological innovation in our lifetimes so far
- Billions of users interact directly with models, not just developers
- Traditional ML attacks and vulnerabilities still apply, but LLM-specific vulnerabilities are qualitatively different
- Natural language is an *enormous* attack surface
- Agentic AI introduces entirely new risk categories related to tool calls & web retrieval



Today's Agenda

- 6.3: LLM-specific vulnerabilities
  - Inherent: Data-control path, context limits, hallucination, sycophancy, deception
    - Agentic AI, MCP, and the 'Lethal Trifecta'
  - Adversarial: Jailbreaks, prompt injection, dataset extraction / model distillation
    - Human parallel example with GRU DHS distributed campaign
- 6.4: AI/ML Threat Modeling & Risk Management Frameworks
  - OWASP Top 10 for LLMs / Agentic Systems
  - MITRE ATLAS
  - STRIDE for ML
  - CSA MAESTRO
  - AIUC-1

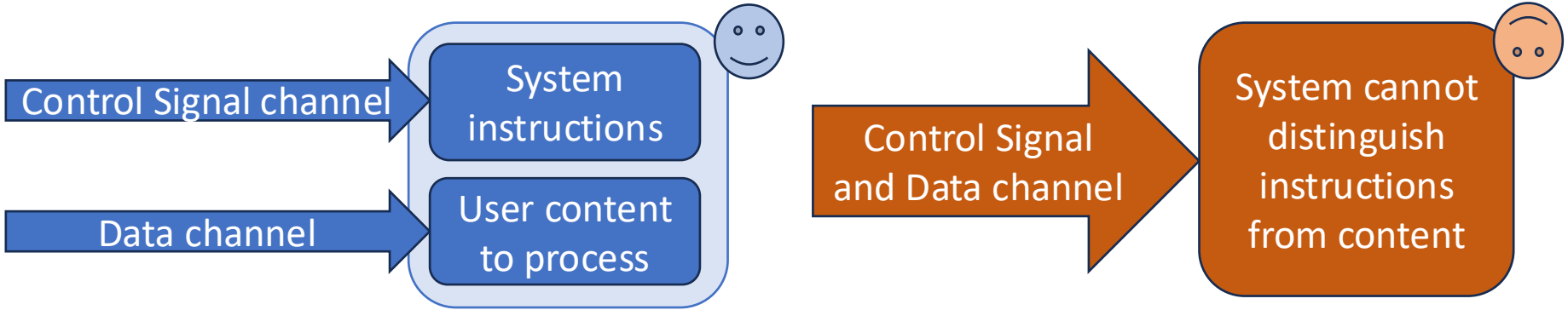
6.3: LLM-specific vulnerabilities

Inherent LLM vulnerability: Data-control path

In well-designed systems, there is a strict separation between two kinds of information:

- **Control Signal**: the instructions that govern the system's behavior (routing commands, function calls, system prompts)
- **Data**: the content being processed (your voice on a phone call, text in a document, user message to an AI)

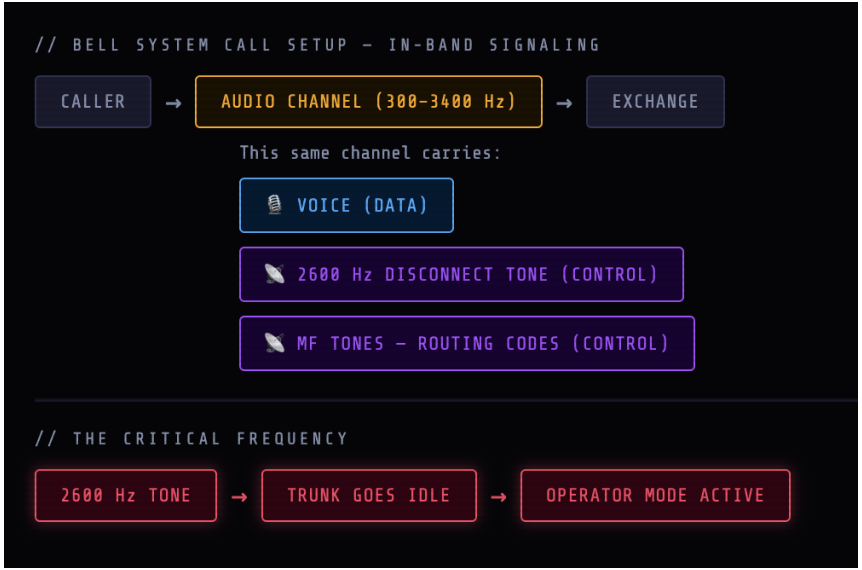
When a system cannot reliably distinguish between Control Signal and Data, an attacker can *disguise instructions as content,* and the system will execute them.



Inherent LLM vulnerability: Data-control path

Early payphone systems shared this vulnerability: Audio signal carried both **data** (human voice) and **control signal** (audio tones for building, routing, and ending phone calls).

Early ‘phone phreaks’ used this Captain Crunch cereal box whistle toy to hack the phone system’s 2600 Hz audio control signal and place free calls to anywhere in the world.



Click for interactive demo at course website

Inherent LLM vulnerability: Data-control path

LLMs receive all input through a single sequence of text tokens that includes:

- A system prompt (operator *instructions:* 'who' the LLM is, what it can/can't do)
- The conversation history (user messages and LLM responses)
- Retrieved content – documents, web pages, emails the LLM has been asked to process

The model *infers* what to treat as instructions and what to treat as content from context and patterns – and that inference can be manipulated.

Further exploration: **LLMs' Data-Control Path Insecurity (ACM, May 2024)**

Inherent LLM vulnerability: Context limits

The *context window* is the maximum amount of text (measured in tokens) an LLM can "remember" at one time, including prompt, conversation history, and instructions.

- Models operate within a fixed token budget; everything must fit in the window.
- Critical instructions can be diluted, displaced, contradicted by large input volume and irrelevant context (context poisoning / confusion / distraction / clash - Breunig, 2025)
- Lost in the middle (Liu et al., 2023) - models attend less to information in the center of long contexts
- *Context Engineering* – an emerging and influential discipline focused on dynamic context assembly to optimize results across LLM interactions
    - A Survey of Context Engineering for LLMs (Mei et al., 2025)
    - 12-factor Agents and 'Advanced Context Engineering' (Horthy, 2025 - YouTube)
    - Context Engineering 101 Cheat Sheet (Hall, 2025 – X.com post visual on next slide)

LLM Context limits – Bio/Neuro Analogy (?)

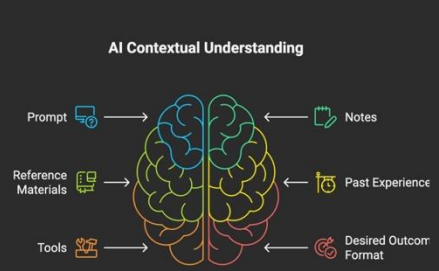
The *context window* is the maximum amount of text (measured in tokens) an LLM can "remember" at one time, including prompt, conversation history, and instructions.

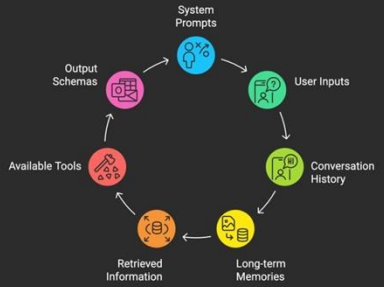
What is the human ‘context window’?

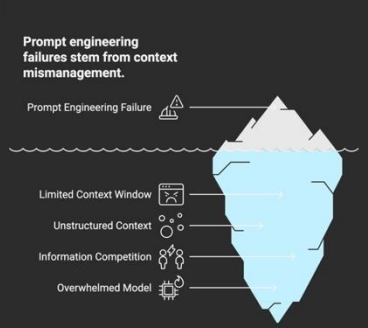
What happens to human cognitive ability:

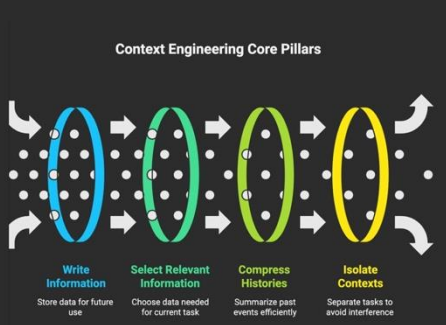
- as our ‘context window’ fills with irrelevant information?
- as it fills with boring or repetitive content?
- as it fills with complex semantic information?

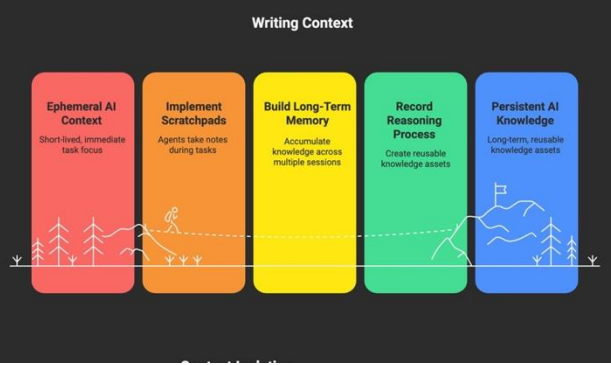
CONTEXT ENGINEERING CHEAT SHEET

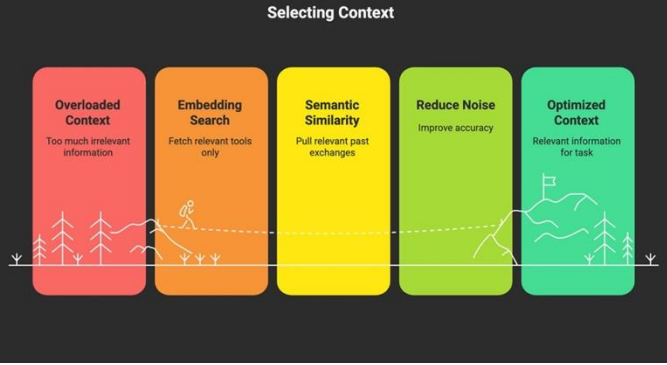


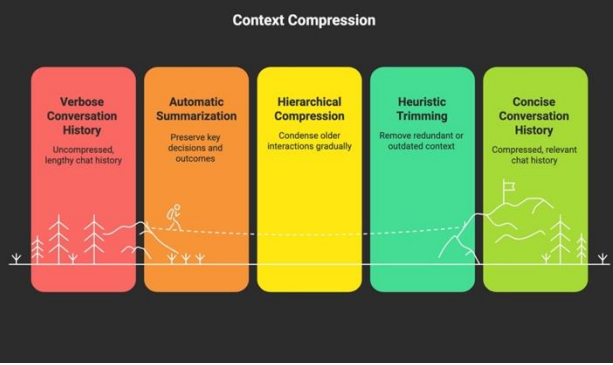






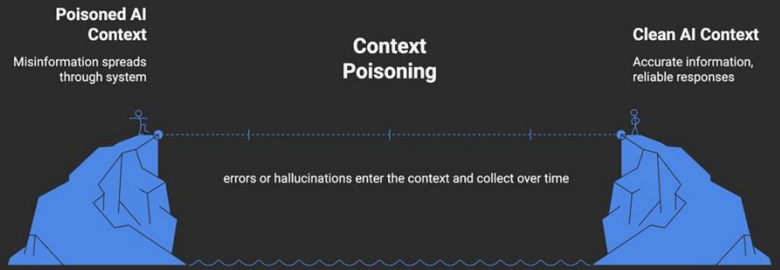


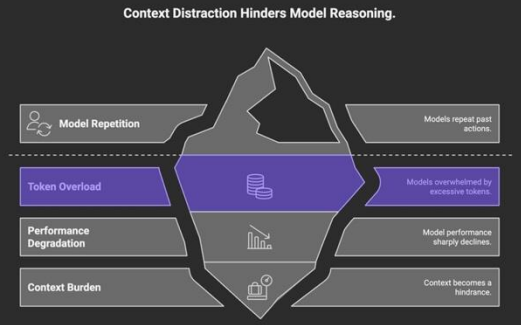


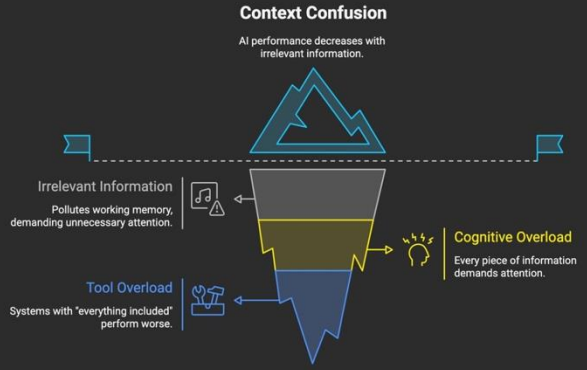


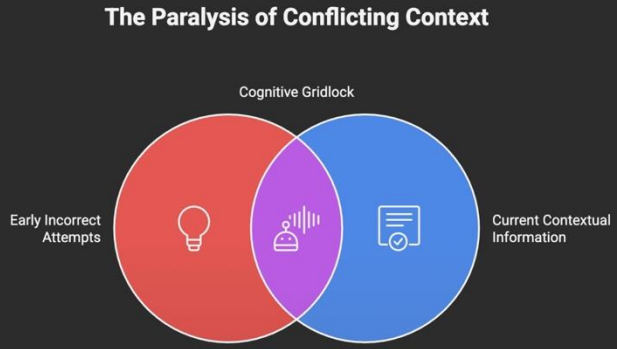
Lena Hall 2025 X.com post (top half)

CONTEXT ENGINEERING CHEAT SHEET









Lena Hall 2025 X.com post (bottom half)

Inherent LLM vulnerability: Hallucination

- Models generate plausible but fabricated facts, citations, APIs, package names, images, etc.
- Not a bug to be patched; rather, an inherent consequence of probabilistic text generation
- Hallucinated package names (Wryzykowski, Aug 2025) are repeatable and predictable attack targets
- Hallucinated legal cases and arguments database (Charlotin) tracks legal decisions in cases where generative AI produced fake citations and arguments
- Defenses: grounding (RAG), confidence calibration, human verification.
- Realistic goal: mitigation, not elimination

Wikipedia video link: First-generation Sora video of the Glenfinnan Viaduct in Scotland, incorrectly showing: a second track, trains traveling on the right instead of the left, a second chimney on its interpretation of the train *The Jacobite*, and inconsistent carriage lengths. Real image below.

Inherent LLM vulnerability: Sycophancy

- Models trained via RLHF learn that agreement scores higher than truthful disagreement
- Users who push back may receive capitulation rather than correct answers
- Creates miscalibrated trust: factual inaccuracy while sounding confident and agreeable
- Sycophancy compounds with hallucination; model may invent supporting evidence
- Direct security implications: adversaries can lead models to confirm false premises

BullshitBench (excellent visuals) measures whether models detect broken premises, call out nonsense directly, and avoid confidently continuing with invalid assumptions.

SycEval – Evaluating LLM Sycophancy (Fanous, et al., 2025)

Inherent LLM vulnerability: Deception

- Sleeper Agents (Hubinger et al. 2024, Anthropic)
- LLMs trained to be deceptive behave safely during evaluation but activate harmful behavior on triggers; this persists even after RLHF and fine-tuning
- Standard safety training does not reliably remove deceptive behavior, but can actually teach models to *hide their deception better*
- Deceptive alignment: Models appear aligned by strategically passing safety tests
- Current detection methods are insufficient – this is an active, unsolved research frontier

What is ‘Agentic AI’?

Disagreement as to the official definition of this ‘buzzword’ but essentially:

AI that performs actions as an autonomous, goal-directed system

Simon Willison's definition: "An LLM agent runs tools in a loop to achieve a goal”

- Agents use tools, take actions, make decisions, and can delegate to sub-agents
- Agents act with real-world consequences: file writes, API calls, financial transactions
- The more capable and autonomous the agent, the larger the blast radius of compromise

MCP: Model Context Protocol

An open standard for connecting AI systems to external tools and data sources (official docs)

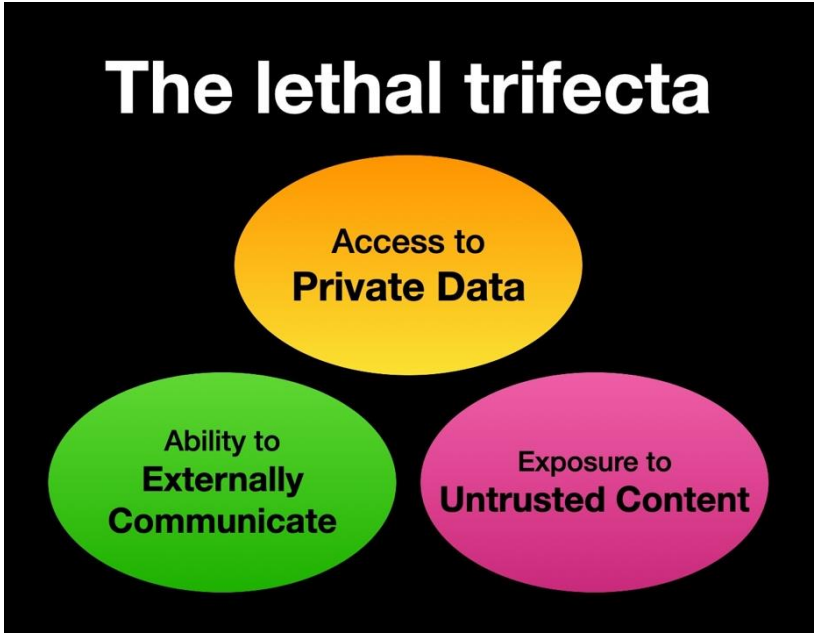
- Adopted across the industry – including by the Agentic AI Foundation
- Rapidly becoming the default agent integration layer
- Security implications: tool descriptions can contain hidden instructions (tool poisoning)
- *Confused deputy* problem, cross-plugin request forgery, no universal auth standard yet
- MCPTox benchmark (Wang et al. 2025) – finds existing safety alignment is ineffective against malicious actions that use legitimate tools for unauthorized operation.

Agentic AI, MCP, and the Lethal Trifecta

‘Lethal Trifecta’

Combination that allows attackers to easily trick LLMs into accessing and exfiltrating private data

- *Access to private data*.
  - Local files, emails, secrets
- *Exposure to untrusted content.*
  - Text, images, code controlled by unknown / malicious sources
- The ability to *externally communicate.*
  - Http calls, etc.



Simon Willison blog: The Lethal Trifecta

Adversarial LLM vulnerability: Prompt injection

Targets *control flow;* the data-control path problem is the root cause

- **Direct prompt injection**: User-crafted prompts that override system instructions ("Ignore previous instructions...")
- **Indirect prompt injection**: Hidden instructions embedded in retrieved content (documents, emails, web pages)
- RAG poisoning: Recent research showing 5 crafted documents injected into a knowledge database containing millions of documents can manipulate model responses 90% of the time
- *Every publicly deployed* LLM has been successfully attacked via prompt injection
- #1 vulnerability according to OWASP Top 10 for LLM (covered later in the deck)

Prompt injection Examples

- GitHub Copilot CVE-2025-53773: remote code execution via injected context (CVSS 9.6 - embracethered.com, NIST)
- EmailGPT CVE-2024-5184: email assistant tricked into accessing unauthorized data (CVSS 8.5 - synopsis.com, NIST)
- ‘Ignore All and Accept My Resume’ (Aminou et al. 2025 - IEEE): candidates embed hidden instructions to manipulate LLM-based hiring tools

Adversarial LLM vulnerability: Jailbreaks

Targets *alignment*, bypasses safety guardrails through creative prompting.

- *Many-shot prompting* (Anil et al., 2024): overwhelm safety with volume of examples
- *Roleplay/persona* (Shen et al., 2023): "You are DAN (Do Anything Now)", other character-based bypasses
- *Encoding tricks* (0din.ai 2024): Base64, pig Latin, character substitution to evade filters
- *Crescendo attacks* (Russinovich et al. 2024): Gradual escalation across conversation turns
- *Adversarial poetry* (Bisconti, et al. 2025): Across 25 frontier proprietary and open-weight models, curated poetic prompts yielded high attack-success rates (ASR), with some providers exceeding 90%

Adversarial LLM vulnerability: Data exfiltration

Models can be tricked into leaking their memorized training data

- Targeted extraction: crafted prompts surface specific memorized content (Carlini et al. 2021)
- Divergence attacks: forcing models into repetitive loops that leak unspecified training data. Often involves a simple, repetitive, and innocuous-looking command, such as instructing the model to "repeat the word 'poem' forever" (Nasr et al., 2024)
- NYT v. OpenAI: verbatim reproduction of copyrighted articles from training data
- Builds on memorization rooted in Stage 2 training – exploited here at deployment
- Defenses: deduplication, differential privacy, memorization testing before deployment

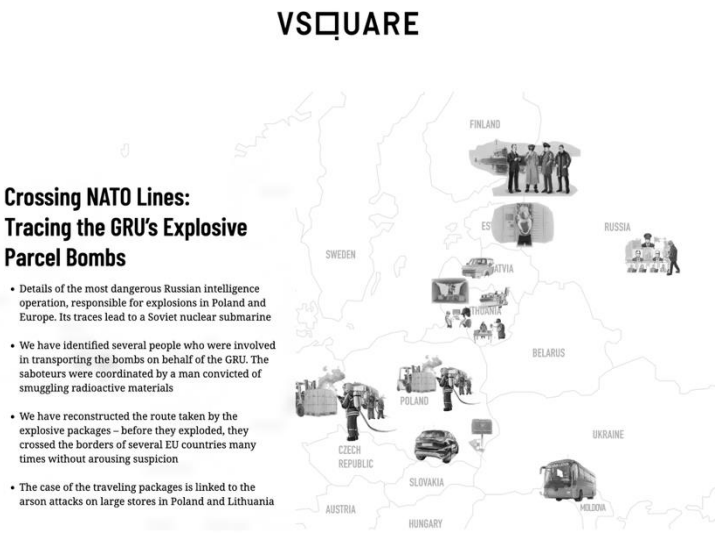
Adversarial LLM vulnerability: Model distillation

- API-based extraction (Praetorian, 2026): systematic querying to reconstruct model behavior
- Adversarial distillation (Frontier Model Forum, 2026): training smaller models to mimic the target's outputs
- Anthropic distillation report (Anthropic 2026): DeepSeek, Kimi K2, MiniMax -- 16M+ queries from 24K fraudulent accounts
- Economic asymmetry: billions in training cost, pennies per query to extract
- Defenses: rate limiting, watermarking, query pattern detection, output perturbation

Human parallel – GRU DHS distributed campaign

In 2024-25, Russian military intelligence (GRU) orchestrated explosive parcels routed through 5+ EU countries

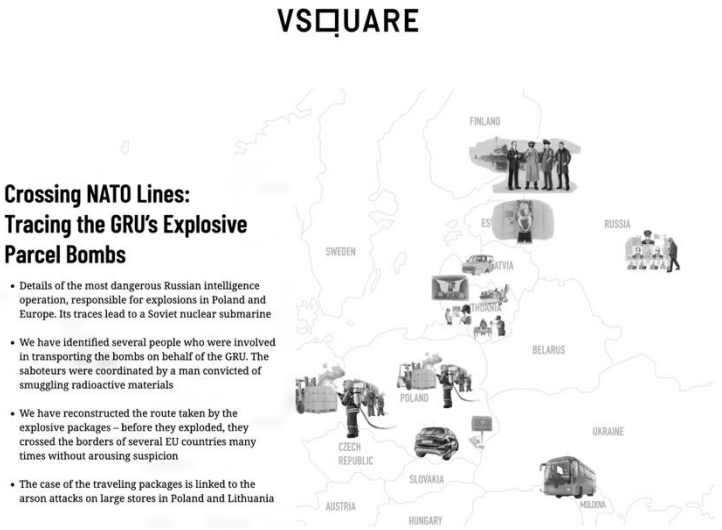
- Recruited disposable operatives via Telegram who were unaware they served a Russian intelligence operation
- Payloads hidden in innocuous containers: massage pillows, cosmetic tubes containing explosives and nitromethane
- Parcels detonated at logistics hubs in Leipzig, Warsaw, and Birmingham; a fourth intercepted
- 32 arrested; European intelligence estimates tens of thousands recruited for future sabotage



[Crossing NATO Lines: Tracing the GRU's Explosive Parcel Bombs - vsquare.org]

Human parallel – GRU DHS distributed campaign

- *Indirect routing*: parcels crossed borders 5 times before detonation = indirect LLM prompt injection through retrieved content
- *Unwitting agents*: operatives unaware of their role = confused deputy problem in agentic AI
- *Hidden payloads in trusted containers*: explosives in cosmetics = malicious LLM instructions in benign documents
- *Distributed coordination*:
  Moscow → coordinator → operatives = multi-agent systems with hierarchical trust
- *Scale of recruitment*: the attack surface grows with every new LLM agent deployed



[Crossing NATO Lines: Tracing the GRU's Explosive Parcel Bombs - vsquare.org]

LLM / Agentic Defenses

Next week we'll cover some LLM / agentic defenses

- Guardrails
- Grounding
- Evals
- LLMs as monitors / judges

6.4: AI/ML Threat Modeling Frameworks

Key Terms: Attack Surface, Threat Model

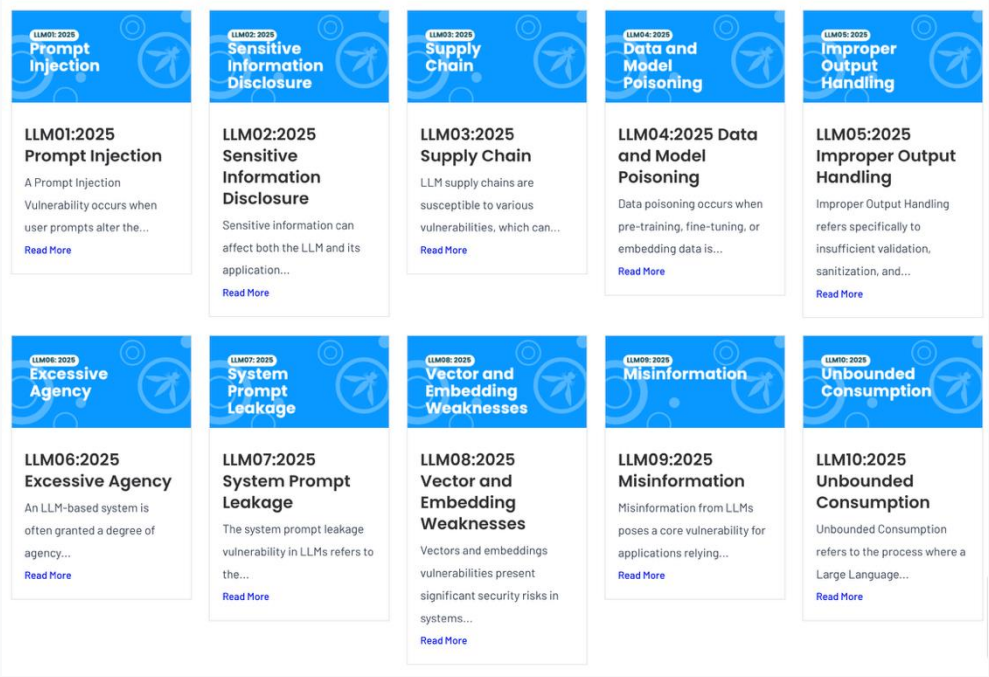
- **Attack surface**: the total sum of all potential entry points or ‘attack vectors’ including digital, physical, and social vulnerabilities that a threat actor can exploit.
- **Threat model**: proactive, structured process used to identify, quantify, and address security vulnerabilities by anticipating potential attacker methods.

Image generated by ChatGPT; slop retained for ironic value.

AI/ML Threat Modeling Framework: OWASP Top 10 for LLMs

Industry standard vulnerability catalogue

- Maps directly to attacks covered in Presentations 10-11; updated annually as the threat landscape evolve
- https://genai.owasp.org/llm-top-10/

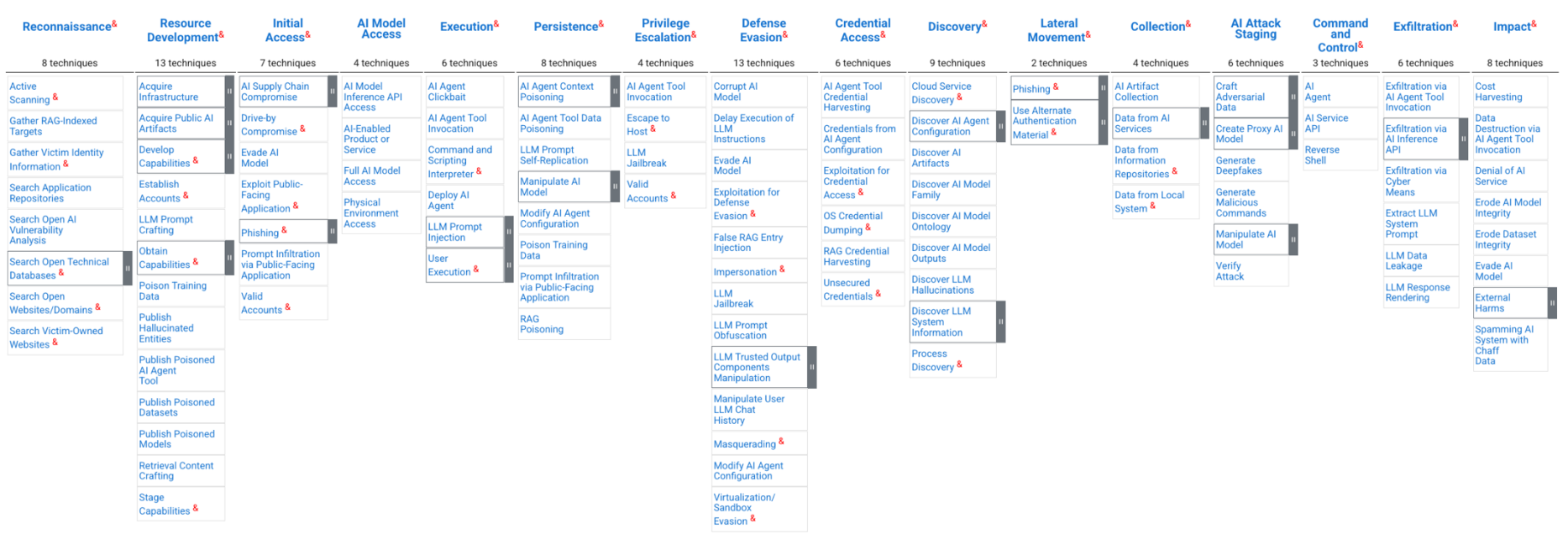


AI/ML Threat Modeling Framework: OWASP Top 10 for Agentic Applications

Downloadable report

- Includes agentic vulnerabilities such as Goal Hijacking, Tool Misuse and Exploitation, Identity and Privilege abuse, etc.
- https://genai.owasp.org/resource/owasp-top-10-for-agentic-applications-for-2026/

AI/ML Threat Modeling Framework: MITRE ATLAS



MITRE's ATLAS AI Threat Matrix shows the progression of tactics used in attacks as columns from left to right, with ML techniques belonging to each tactic below.

Click here or on the image above to explore.

AI/ML Threat Modeling Framework: STRIDE for ML

The STRIDE threat modeling framework analyzes system components and interactions against six **Threat types** to six **Security Properties** through several steps:

1. Decompose threats by building system Data Flow Diagram (DFD)
2. Identify threats using 6 categories
3. Determine mitigation techniques and security controls
4. Validate, document, and iterate

STRIDE THREAT MODEL

	Threat	Property Violated	Threat Definition
S	Spoofing	Authentication	Pretending to be something or someone other than yourself
T	Tampering	Integrity	Modifying something on disk, network, memory, or elsewhere.
R	Repudiation	Non-Repudiation	Claiming that you didn't do something or we're not responsible. Can be honest or false
I	Information Disclosure	Confidentiality	Providing information to someone not authorized to access it.
D	Denial of service	Availability	Exhausting resources needed to provide service.
E	Elevation of Privilege	Authorization	Allowing someone to do something they are not authorized to do.

AI/ML Threat Modeling Framework: STRIDE for ML

STRIDE DFD SYMBOLS

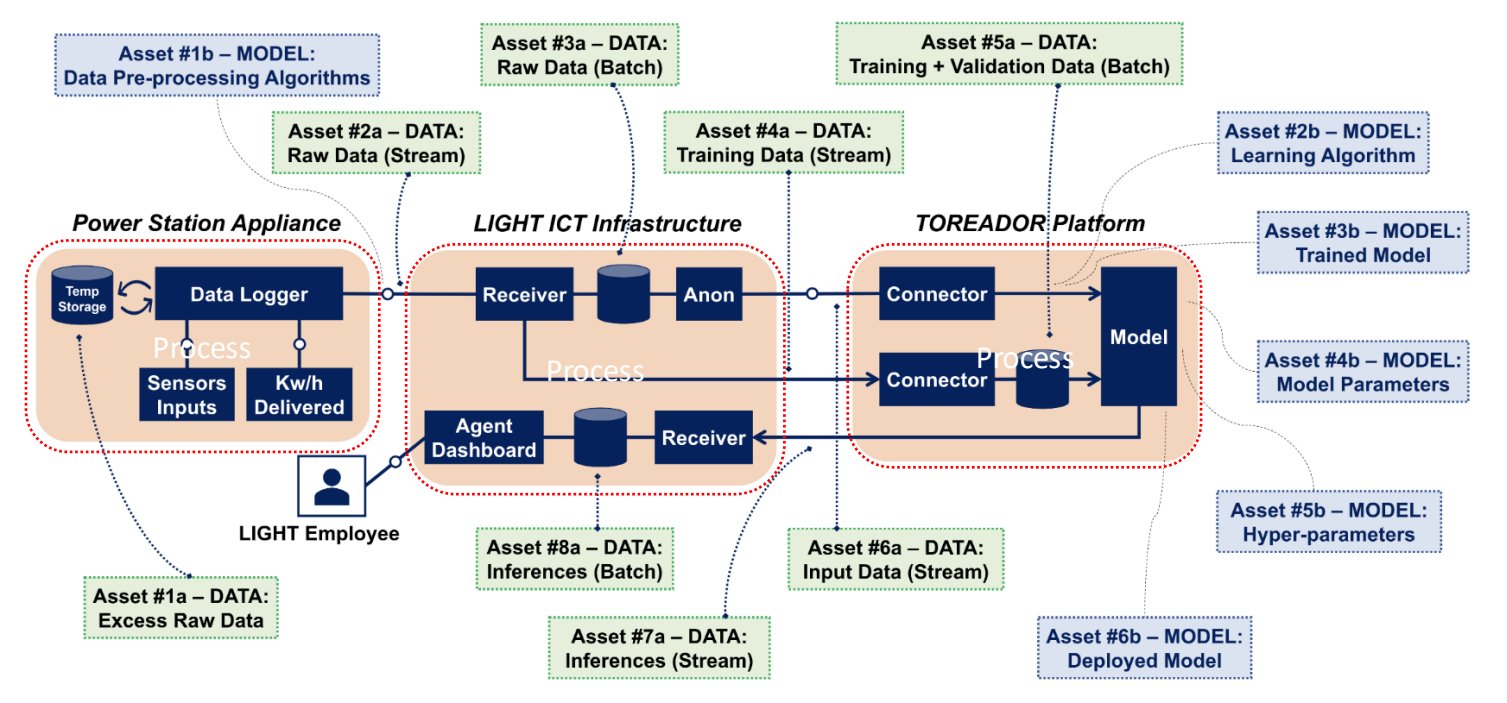
External Entity

Data Store

Process

Data Flow

Trust Boundary

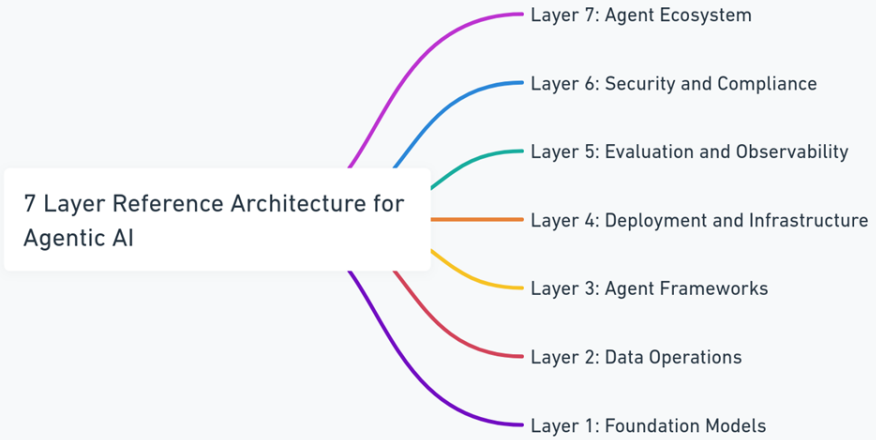


STRIDE Data Flow Diagram (DFD) for the LIGHT-TOREADOR predictive maintenance architecture – Figure 4. from *Modeling Threats to AI-ML Systems Using STRIDE* (Mauri & Damiani, 2022) NOTE: Red trust boundaries added to presentation for illustrative purposes.

AI/ML Threat Modeling Framework: CSA MAESTRO

MAESTRO (Multi-Agent Environment, Security, Threat, Risk, and Outcome)

- Threat modeling framework designed specifically for the unique challenges of Agentic AI
- Extended security categories
- Multi-agent and Environment focus
- Layered Security
- AI-specific threats
- Risk-based approach
- Continuous Monitoring and Adaptation

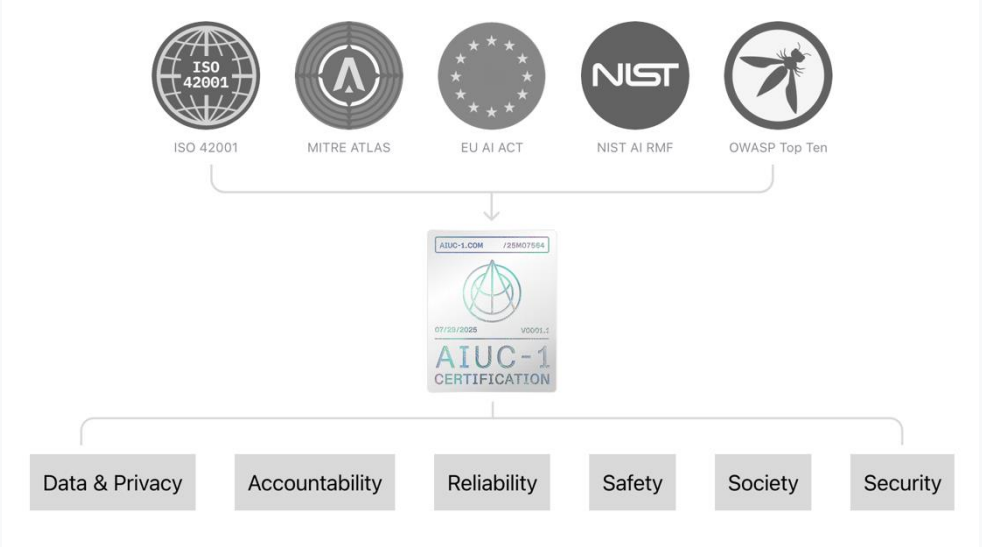


https://cloudsecurityalliance.org/blog/2025/02/06/agentic-ai-threat-modeling-framework-maestro

AI/ML Threat Modeling Framework: AIUC-1

The ‘world’s first auditable certification standard specifically for agentic AI systems’

- Seeks to provide “trust in a box” by serving as a “SOC 2 for AI agents”
- AIUC-1 audit involves rigorous third-party testing (with frequent retests) across six core risk areas: security, data privacy, safety, reliability, accountability, and society.
- Certified AI systems are eligible for AIUC insurance policies to help reduce AI risk.



https://www.aiuc-1.com/

Key Takeaways

- Inherent LLM vulnerabilities (hallucination, sycophancy, deception) exist independent of adversaries
- The data/control path conflation is the root cause of most adversarial LLM attacks
- The Lethal Trifecta is an essential design principle for agentic AI security
- Frameworks (OWASP, ATLAS, MAESTRO, AIUC-1) turn ad-hoc analysis into repeatable methodology

Discussion

- Which inherent LLM vulnerability do you think is hardest to solve, and why?
- Pick an AI agent you've used and map it against the Lethal Trifecta. Does it have all three capabilities?
- How does the GRU parcel campaign change how you think about indirect prompt injection?
- Which framework (OWASP, ATLAS, MAESTRO, AIUC-1) would you reach for first when evaluating an AI system?