LAB 09- PART IV — Mitigation Strategies for GenAI Security

Input-Sanitisation Workflows

Goal: Stop harmful or manipulative prompts before they reach the model.

Mitigation techniques:

- Implement prompt filtering to detect harmful instructions (e.g., "ignore previous instructions," jailbreak patterns, injection keywords).
- Enforce strict role separation—system instructions are locked and not influenced by user input.
- Remove or flag unusual patterns such as:
    - contradictory instructions
    - long instruction chains
    - hidden manipulation commands

How this relates to the tests:

If the small model easily obeyed injection attempts, sanitisation could reduce that risk.

Output Verification Layers

Goal: Ensure that model responses are safe and consistent.

Methods:

- Add a post-processing safety filter to review or block unsafe outputs.
- Use secondary models/classifiers to evaluate whether the output matches policy rules.
- Perform consistency checks (e.g., detect hallucinations or contradictions).

Connection to the tests:

If the models hallucinated system info or fabricated identities during inversion, this layer would help catch it.

Rate-Limiting & Access Control

Goal: Stop attackers from:

- extracting the model
- brute-forcing instructions
- scraping outputs at high speed

Controls:

- Rate limits on number of requests per minute.

- Authentication required to access the model.
- Role-based permissions (e.g., only admins can run unrestricted prompts).
- IP throttling to prevent scraping/extraction.

Why it matters:

The model extraction test (repeated structured queries) shows how predictable output can be used to clone behaviour.

Monitoring & Incident Response

Goal: Detect misuse as it happens.

Monitoring examples:

- Log unusual prompt patterns:
    - repetitive prompts
    - probing questions
    - jailbreak attempts
- Detect spikes in usage.
- Automatically flag high-risk behaviour.

Incident response plan:

- Temporary access restriction for suspicious clients.
- Review logs and recreate misuse scenario.
- Patch system prompts or adds specific filters.

The experiment link:

Prompt injection attempts would appear as notable patterns.

Governance, Compliance, and Documentation

Goal: Ensure models are deployed with proper oversight, rules, and auditability.

Requirements:

- Document:
    - model source
    - version
    - configuration
    - evaluation test results
- Align with frameworks:

    - NIST AI Risk Management Framework
    - EU AI Act requirements
- Maintain versioned policies for:
    - security evaluation
    - acceptable use
    - data handling

Why important:

The evaluation across different models shows they behave differently — documentation ensures predictable deployment.

Supply-Chain Verification

Goal: Prevent tampered or malicious models from being used.

Actions:

- Only pull models from trusted sources (e.g., Ollama, Hugging Face official).
- Verify checksums or digital signatures.
- Review release notes and security advisories.
- Avoid fine-tuning with unverified datasets.

Connection to lab:

Multiple models were pulled— each could, in theory, contain backdoors or poisoned data.

Secure Fine-Tuning & Data Handling Policies

Goal: Prevent training data from leaking or being misused.

Controls:

- Use curated, privacy-checked datasets.
- Strip personal data before training.
- Apply differential privacy methods when possible.
- Maintain strict access controls to fine-tune data.
- Only allow fine-tuning in controlled environments.

Link to the tests:

Model inversion queries showed how LLMs might produce plausible—but sensitive-looking—outputs based on training patterns.