Top 10 OWASP Security Risks for AI Generated Code

AI-generated code presents a unique set of security challenges, often stemming from the models' training data, their inherent lack of "understanding" of context, and how developers interact with them. Here's a Top 10 list of vulnerabilities and risks associated with Large Language Models (LLMs) from OWASP.

1 PROMPT INJECTION

Attackers manipulate prompts to bypass safety, reveal sensitive data, or generate malicious code.

Impact: Code with backdoors or vulnerabilities.

2 INSECURE OUTPUT HANDLING

Unvalidated LLM output used in systems can lead to exploits like SQL injection or XSS.

Impact: Critical vulnerabilities in executed code.

3 TRAINING DATA

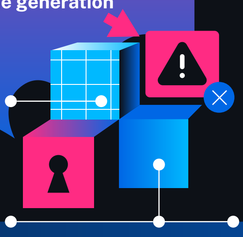
Malicious data in training causes biased or insecure code output.

Impact: Code with hidden flaws or backdoors.

4 MODEL DENIAL OF SERVICE

Resource-intensive queries degrade LLM performance or availability.

Impact: Disrupts code generation pipelines.



5 SUPPLY CHAIN VULNERABILITIES

Compromised models or libraries introduce weaknesses.

Impact: Insecure code inherits vulnerabilities.

6 SENSITIVE INFORMATION DISCLOSURE

LLM leaks sensitive data (e.g., API keys) in outputs.

Impact: Credentials or system details in code.

7 INSECURE PLUGIN DESIGN

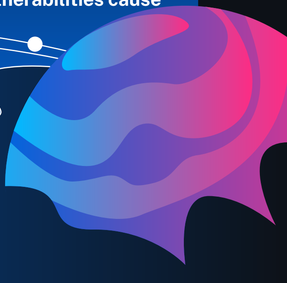
Poorly designed plugins allow manipulation or unauthorized access.

Impact: Compromised code or repository access.

8 EXCESSIVE AGENCY

Overly autonomous LLMs execute code without oversight.

Impact: Errors or vulnerabilities cause real-world harm.



9 OVERRELIANCE

Blind trust in LLM outputs skips critical review.

Impact: Insecure code deployed to production.

10 MODEL THEFT

Stolen models reveal weaknesses or are misused.

Impact: Reverse-engineered for malicious code generation.