# The periodic table of AI security

*Found at: https://owaspai.org/goto/periodictable/*

The table below, created by the OWASP AI Exchange, shows the various threats to AI and the controls you can use against them – all organized by asset, impact and attack surface, with deeplinks to comprehensive coverage at the AI Exchange website with further references to related standards.

Note that general governance controls apply to all threats.

| Asset & Impact | Attack surface with lifecycle | Threat/Risk category | Controls |
|---|---|---|---|
| Model behaviour Integrity | Runtime -Model use (provide input/ read output) | Direct prompt injection | Limit unwanted behavior, Input validation, further controls implemented in the model itself |
| | | Indirect prompt injection | Input validation, Input segregation |
| | | Evasion (e.g. adversarial examples) | Limit unwanted behavior, Monitor, rate limit, model access control plus:<br><br>Detect odd input, detect adversarial input, evasion robust model, train adversarial, input distortion, adversarial robust distillation |
| | Runtime - Break into deployed model | Model poisoning runtime (reprogramming) | Limit unwanted behavior, Runtime model integrity, runtime model input/output integrity |
| | Development - Engineering environment | Model poisoning development time | Limit unwanted behavior, Development environment security, data segregation, federated learning, supply chain management plus:<br><br>model ensemble |
| | | Data poisoning of train/finetune data | Limit unwanted behavior, Development environment security, data segregation, federated learning, supply chain management plus:<br><br>model ensemble plus:<br><br>More training data, data quality control, train data distortion, poison robust model |
| | Development - Supply chain | Model/data poisoning in supply chain | Limit unwanted behavior,<br>Supplier: Development environment security, data segregation, federated learning<br><br>Producer: supply chain management plus:<br><br>model ensemble |
| Training data Confidentiality | Runtime - Model use | Data disclosure in model output | Sensitive data limitation (data minimize, short retain, obfuscate training data) plus:<br><br>Monitor, rate limit, model access control plus:<br><br>Filter sensitive model output |
| | | Model inversion / Membership inference | Sensitive data limitation (data minimize, short retain, obfuscate training data) plus:<br><br>Monitor, rate limit, model access control plus:<br><br>Obscure confidence, Small model |
| | Development - Engineering environment | Training data leaks | Sensitive data limitation (data minimize, short retain, obfuscate training data) plus:<br><br>Development environment security, data segregation, federated learning |
| Model confidentiality | Runtime - Model use | Model theft through use (input-output harvesting) | Monitor, rate limit, model access control |
| | Runtime - Break into deployed model | Direct model theft runtime | Runtime model confidentiality, Model obfuscation |
| | Development - Engineering environment | Model theft development-time | Development environment security, data segregation, federated learning |
| Model behaviour Availability | Model use | Denial of model service (model resource depletion) | Monitor, rate limit, model access control plus:<br><br>Dos input validation, limit resources |
| Model input data Confidentialiy | Runtime - All IT | Model input leak | Model input confidentiality |
| Any asset, CIA | Runtime - All IT | Model output contains injection | Encode model output |
| Any asset, CIA | Runtime - All IT | Conventional runtime security attack on conventional asset | Conventional runtime security controls |
| Any asset, CIA | Runtime - All IT | Conventional attack on conventional supply chain | Conventional supply chain management controls |