AI Security Overview

This page provides an introduction, high-over views of threats and controls, step-by-step risk analysis, and discussion of cross-cutting concerns. The next pages go deeper into threats and controls.

Other pages

Topics on this page

0. AI Security Overview

Table of contents

> Category: discussion
> Permalink: https://owaspai.org/goto/toc/

About the AI Exchange

> **Category: discussion**
> **Permalink: https://owaspai.org/goto/about/**

If you want to jump right into the content, head on to the **Table of contents** or **How to use this document**.

Summary

Welcome to the go-to comprehensive resource for AI security & privacy - over 300 pages of practical advice and references on protecting AI, and data-centric systems from threats - where AI consists of ALL AI: Analytical AI, Discriminative AI, Generative AI and heuristic systems. This content serves as key bookmark for practitioners, and is contributed actively and substantially to international standards such as ISO/IEC and the AI Act through official standard partnerships. Through broad collaboration with key institutes and SDOs, the *Exchange* represents the consensus on AI security and privacy.

Details

The OWASP AI Exchange has open sourced the global discussion on the security and privacy of AI and data-centric systems. It is an open collaborative OWASP Flagship project to advance the development of AI security & privacy standards, by providing a comprehensive framework of AI threats, controls, and related best practices. Through a unique official liaison partnership, this content is feeding into standards for the EU AI Act (70 pages contributed), ISO/IEC 27090 (AI security, 70 pages contributed), ISO/IEC 27091 (AI privacy), and **OpenCRE** - which we are currently preparing to provide the AI Exchange content through the security chatbot **OpenCRE-Chat**.

Data-centric systems can be divided into AI systems and 'big data' systems that don't have an AI model (e.g., data warehousing, BI, reporting, big data) to which many of the threats and controls in the AI Exchange are relevant: data poisoning, data supply chain management, data pipeline security, etc.

Security here means preventing unauthorized access, use, disclosure, disruption, modification, or destruction. Modification includes manipulating the behaviour of an AI model in unwanted ways.

Our **mission** is to be the go-to resource for security & privacy practitioners for AI and data-centric systems,

to foster alignment, and drive collaboration among initiatives. By doing so, we provide a safe, open, and independent place to find and share insights for everyone. Follow **AI Exchange at LinkedIn**.

How it works

The AI Exchange is displayed here at **owaspai.org** and edited using a **GitHub repository** (see the links *Edit on Github*). It is an **open-source living publication** for the worldwide exchange of AI security & privacy expertise. It is structured as one coherent resource consisting of several sections under 'content', each represented by a page on this website.

This material is evolving constantly through open source continuous delivery. The authors group consists of over 70 carefully selected experts (researchers, practitioners, vendors, data scientists, etc.) and other people in the community are welcome to provide input too. See the **contribute page**.

OWASP AI Exchange by The AI security community is marked with **CC0 1.0** meaning you can use any part freely without copyright and without attribution. If possible, it would be nice if the OWASP AI Exchange is credited and/or linked to, for readers to find more information.

Who is this for

The Exchange is for practitioners in security, privacy, engineering, testing, governance, and for end users in an organization - anyone interested in the security and privacy of AI systems. The goal is to make the material as easy as possible to access. Using the **Risk analysis section** you can quickly narrow down the issues that matter to your situation, whether you are a large equipment manufacturer designing an AI medical device, or a small travel agency using a chatbot for HR purposes.

History

The AI Exchange was founded in 2022 by **Rob van der Veer** - bridge builder for security standards, Chief AI Officer at **Software Improvement Group**, with 33 years of experience in AI & security, lead author of ISO/IEC 5338 on AI lifecycle, founding father of OpenCRE, and currently working in ISO/IEC 27090, ISO/IEC 27091 and the EU AI act in CEN/CENELEC, where he was elected co-editor by the EU member states.

The project started out as the 'AI security and privacy guide' in October 22 and was rebranded a year later as 'AI Exchange' to highlight the element of global collaboration. In March 2025 the AI Exchange was awarded the status of 'OWASP Flagship project' because of its critical importance, together with the **'GenAI Security Project'**.

The AI Exchange is trusted by industry giants

Dimitri van Zantvliet, Director Cybersecurity, Dutch Railways:

> **"A risk-based, context-aware approach—like the one OWASP Exchange champions—not only supports the responsible use of AI, but ensures that real threats are mitigated without burdening engineers with irrelevant checklists. We need standards written by those who build and defend these systems every day."**

Sri Manda, Chief Security & Trust Officer at Peloton Interactive:

> **"AI regulation is critical for protecting safety and security, and for creating a level playing field for vendors. The challenge is to remove legal uncertainty by making standards really clear, and to avoid unnecessary requirements by building in flexible compliance. I'm very happy to see that OWASP Exchange has taken on these challenges by bringing the security community to the table to ensure we get standards that work."**

Prateek Kalasannavar, Staff AI Security Engineer, Lenovo:

> **"At Lenovo, we're operationalizing AI product security at scale, from embedded inference on devices to large-scale cloud-hosted models. OWASP AI Exchange serves as a vital anchor for mapping evolving attack surfaces, codifying AI-specific testing methodologies, and driving community-aligned standards for AI risk mitigation. It bridges the gap between theory and engineering."**

Mission/vision

The mission of the AI Exchange is to enable people to find and use information to ensure that AI systems are secure and privacy preserving.

The vision of the AI Exchange is that the main challenge for people is to find the right information and then understand it so it can be turned into action. One of the underlying issues is the complexity, inconsistency, fragmentation and incompleteness of the standards and guideline landscape - with issues of quality and being outdated - caused by the general lack of expertise in AI security in the industry. What resource to use?

The AI Exchange achieves:

- AUTHORITATIVE - active alignment with other resources through careful analysis and through close collaboration - particularly through substantial contribution to leading international standards at ISO/IEC and the AI Act - making sure the AI Exchange represents consensus.
- OPEN - Anybody that wants to, can contribute to the AI Exchange body of knowlesge, with strong quality assurance, including a screening process for Authors.
- FREE - Anybody that wants to, use can use it in any way. Free of copyright and attribution.
- COVERAGE - comprehensive guidance instead of a selected set of issues (like a top 10 which is more for awareness) - and about all AI and data-intensive systems. AI is much more than Generative AI.
- UNIFIED - a coherent resource instead of a fragmented set of disconnected separate resources.
- CLEAR - clear explanation, including also the why and how and not just the what.
- LINKED - referring to various other sources instead of complex text that incorrectly suggests it is complete. This makes the Exchange the place to start
- EVOLVING - continuous updates instead of occasional publications

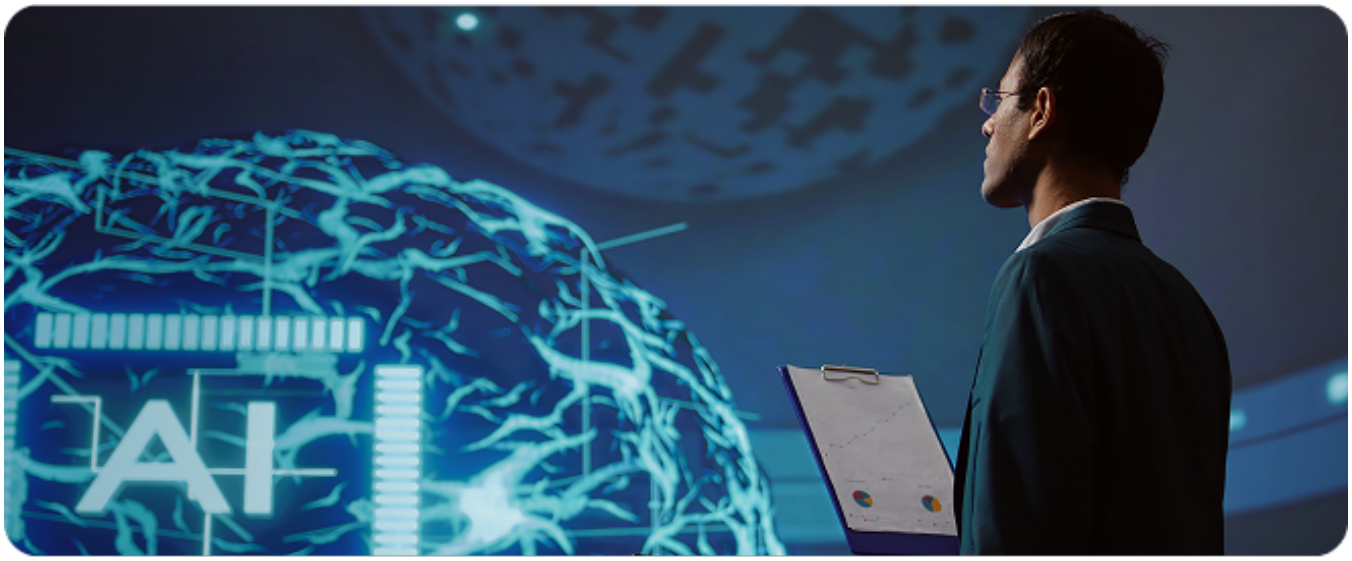
- EVOLVING - continuous updates instead of occasional publications.

All aspects above make the Exchange the go-to resource for practitioners, users, and training institutes - effectively and informally making the AI Exchange the standard in AI security.

NOTE: Producing and continuously updating a comprehensive and coherent quality resource requires a strong coordinated approach. It is much harder than an approach of every person for themselves. But necessary.

Relevant OWASP AI initiatives

> **Category: discussion**
> **Permalink: https://owaspai.org/goto/aiatowasp/**



In short, the two flagship OWASP AI projects:

- The **OWASP AI Exchange** is a comprehensive core framework of threats, controls and related best practices for all AI, actively aligned with international standards and feeding into them. It covers all types of AI, and next to security it discusses privacy as well.
- The **OWASP GenAI Security Project** is a growing collection of documents on the security of Generative AI, covering a wide range of topics including the LLM top 10.

Here's more information on AI at OWASP:

- If you want to **ensure security or privacy of your AI or data-centric system** (GenAI or not), or want to know where AI security standardisation is going, you can use the **AI Exchange**, and from there you will be referred to relevant further material (including GenAI security project material) where necessary.
- If you want to get a **quick overview** of key security concerns for Large Language Models

- If you want to get a **quick overview** of key security concerns for Large Language Models, check out the **LLM top 10 of the GenAI project**. Please know that it is not complete, intentionally - for example it does not include the security of prompts.
- For **any specific topic** around Generative AI security, check the **GenAI security project** or the **AI Exchange references**.

Some more details on the projects:

- **The OWASP AI Exchange(this work)** is the go-to single resource for AI security & privacy - over 200 pages of practical advice and references on protecting AI, and data-centric systems from threats - where AI consists of Analytical AI, Discriminative AI, Generative AI and heuristic systems. This content serves as a key bookmark for practitioners, and is contributed actively and substantially to international standards such as ISO/IEC and the AI Act through official standard partnerships.
- The **OWASP GenAI Security Project** is an umbrella project of various initiatives that publish documents on Generative AI security, including the LLM AI Security & Governance Checklist and the LLM top 10 - featuring the most severe security risks of Large Language Models.
- **OpenCRE.org** has been established under the OWASP Integration standards project(from the *Project wayfinder*) and holds a catalog of common requirements across various security standards inside and outside of OWASP. OpenCRE will link AI security controls soon.

When comparing the AI Exchange with the GenAI Security Project, the Exchange:

- feeds straight into international standards
- is about all AI and data centric systems instead of just Generative AI
- is delivered as a single resource instead of a collection of documents
- is updated continuously instead of published at specific times
- is focusing on a framework of threats, controls, and related practices, making it more technical-oriented, whereas the GenAI project covers a broader range of aspects
- also covers AI privacy
- is offered completely free of copyright and attribution

How to organize AI Security

> **Category: discussion**
> Permalink: https://owaspai.org/docs/ai_security_overview/

> Permalink: https://owaspai.org/goto/organize/

Organizations: start here!
While Artificial Intelligence (AI) offers tremendous opportunities, it also brings new risks including security threats. It is therefore imperative to approach AI applications with a clear understanding of potential threats and the controls against them. AI can only prosper if we can trust it.

The five steps - G.U.A.R.D - to organize AI security as an organization are:

1. **Govern**
   Start implementing general AI Governance so the organization can manage AI: know where it is applied, what people's responsibilities are, establish policies, do impact assessment, arrange **compliance**, organize **education**, etcetera. See **#AI Program** for guidance, including a quickstart. This is a general AI management process - not just security.
2. **Understand**
   - Based on the inventory of your applications of AI and AI ideas, understand which threats apply, using the decision tree in the **risk analysis section**.
   - Then make sure engineers and security professionals understand those relevant threats and their contols, using the guidance of the relevant **threat sections** and the corresponding **process controls and technical controls**.
   - Use the courses and resources in the **references section** to support the understanding.
   - Distinguish between controls that your organization has to implement, and those that are the responsbility of your supplier. Make the latter category part of your [supply

chain management])(/goto/supplychainmanage/).

3. **Adapt**
   - **Adapt your security practices** to include AI security assets, threats and controls from this document.
   - Adapt your threat modelling to include the **AI security threat model** approach and do cross-team threat modelling, involving all engineers.
   - Adapt your testing to include **AI-specific security testing**.
   - Adapt your supply chain management to include **data, model, and hosting management** and to make sure that your suppliers are taking care of the identified threats.
   - If you develop AI systems (even if you don't train your own models): Adapt your **software development practices** and **secure development program** to involve AI engineering activities.

4. **Reduce**
   Reduce potential impact by **minimizing or obfuscating sensitive data** and **limiting the impact of unwanted behaviour** (e.g., managing privileges, guardrails, human oversight etc. Basically: apply Murphy's law.
5. **Demonstrate**
   Establish evidence of responsible AI security through transparency, testing, documentation, and communication. Prove to management, regulators, and clients that your AI systems are under control and that the applied safeguards work as intended.

And finally: think before you build an AI solution. AI can have fantastic benefits, but it always needs to be balanced with risks. Securing AI is typically harder than securing non-AI systems, first because it's relatively new, but also because there is a level of uncertainty in all data-driven technology. For example in the case of LLMs, we are dealing with the fluidity of natural language. LLMs essentially offer an unstable, undocumented interface with an unclear set of policies. That means that security measures applied to AI often cannot offer security properties to a standard you might be used to with other software. Consider whether AI is the appropriate technology choice for the problem you are trying to solve. Removing an unnecessary AI component eliminates all AI-related risks.

How to use this Document

> **Category: discussion**
> **Permalink: https://owaspai.org/goto/document/**

The AI Exchange is a single coherent resource on the security and privacy of AI systems, presented on this website, divided over several pages - containing threats, controls, guidelines, tests and references.

Ways to start, depending on your need:

- **Learn more what the AI Exchange is**:
  See **About**
- **Start AI security as organization**:
  See **How to organize AI security** for the key steps to get started as organization.
- **Start AI security as individual**:
  See 'learn/lookup' below to familiarize yourself with the threats and controls or look in the **references section** for a large table with training material.
- **Secure a system**:
  If you want your **AI system to be secure**, start with **risk analysis** to guide you through a number of questions, resulting in the threats that apply. And when you click on those threats you'll find the controls (countermeasures) to check for, or to implement.
- **Learn / look up**:
  - For the short story with the main insights in what is special about AI security: see the **AI Exchange essentials**.
  - Ask AI an AI security/privacy question based on the content of the Exchange: **here** (requires Google account).
  - To see a general overview and discussion of all **threats** from different angles, check the **AI threat model** or the **AI security matrix**. In case you know the threat you need to protect against, find it in the overview of your choice and click to get more information and how to protect against it.
  - To find out what to do against a specific threat, check the **controls overview** or the **periodic table** to find the right **controls**.
  - To understand what controls to apply in different deployment models: have a look at the **section on ready-made models**.
  - To learn about **privacy** of AI systems, check **the privacy section**.
  - Agentic AI aspects are covered throughout all content, with a specific section **here**.
  - To look up a specific topic, use the search function or the **index**.
  - Looking for more information, or training material: see the **references**.

- **Test**:
  If you want to **test** the security of AI systems with tools, go to **the testing page**.

The AI exchange covers both heuristic artificial intelligence (e.g., expert systems) and machine learning. This means that when we talk about an AI system, it can for example be a Large Language Model, a linear regression function, a rule-based system, or a lookup table based on statistics. Throughout this document, it is made clear which threats and controls play a role and when.

The structure
You can see the high-level structure on the **main page**. On larger screens you can see the structure of pages on the left sidebar and the structure within the current page on the right. On smaller screens you can view these structures through the menu. There is also a section with the most important topics in a **Table of contents**.

The main structure is made of the following pages:
(0) **AI security overview - this page**, contains an overview of AI security and discussions of various topics.
(1) **General controls, such as AI governance** (2) **Threats through use, such as evasion attacks** (3) **Development-time threats, such as data poisoning** (4) **Runtime security threats, such as insecure output** (5) **AI security testing** (6) **AI privacy** (7) **References**

This page (AI security overview) will continue with discussions about:

- A high-level overview of threats
- Various overviews of threats and controls: the matrix, the periodic table, and the navigator
- Risk analysis to select relevant threats and controls
- Various other topics: heuristic systems, responsible AI, generative AI, the NCSC/CISA guidelines, and copyright

---

AI security essentials

> **Category: discussion**
> **Permalink: https://owaspai.org/goto/essentials/**

This section discusses the essentials of AI security. It serves as THE starting point to understand the bigger picture.
What makes AI special when it comes to security? Well, it deals with a new set of threats and therefore requires new controls. Let's go through them.

New threats (overview **here**):

1. **Model input threats**:
    - **Evasion**: Misleading a model by crafting data to force wrong decisions
    - **Prompt injection**: Misleading a model by crafting instructions to manipulate behaviour

behaviour

- **Extracting data from the model**: training data, augmentation data, or input
- **Extracting of the model itself** by querying the model
- **Resource exhaustion** through use

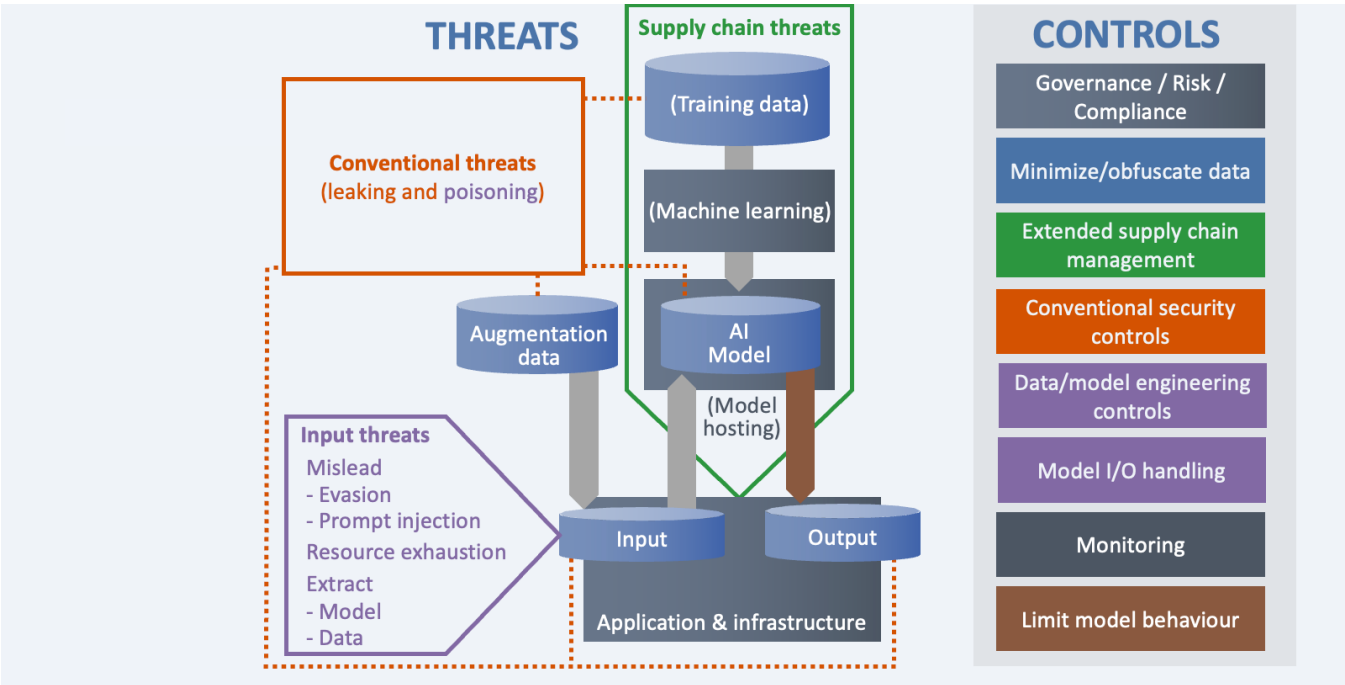
2. **New suppliers** introduce threats of corrupted external **data**, **models**, and **model hosting**
3. **New AI assets** with conventional threats, notably:
    - Training data / augmentation data - can leak and **poisoning** this data manipulates model behaviour
    - Model - can suffer from **leaking during development** or **leaking during runtime** and when it comes to ingegrity: from **poisoning during development** or **poisoning during runtime**
    - Input - can **leak**
    - Output - can contain **injection attacks**

New controls (overview **here**):

- Extend existing **Governance**, **Risk** and **Compliance** - in order to secure AI, you need overview, analysis, policy, training, and responsibilities
- Extend existing **conventional security controls** to protect the AI-specific assets
- Extend **Supply chain management** to incorporate obtaining data, models, and hosting
- Specific **AI engineer controls**, to work against poisoning and model input attacks - next to conventional controls. This category is divided into **Data/model engineering** during development and **Model I/O handling** for runtime filtering, stopping or alerting to suspicious input or output. It is typically the territory of AI experts e.g. data scientists with elements from mathematics, statistics, linguistics and machine learning.
- **Monitoring** of model performance and inference - extending model I/O handing and overlooking general usage of the AI system
- **Impact limitation controls** (because of zero model trust: assume a model can be misled, make mistakes, or leak data):
    - **Minimize or obfuscate sensitive data**
    - **Limit model behaviour** (e.g., **oversight**, **least model privilege**, and **model alignment**)

(*) Note: Attackers that have a similar model (or a copy) can typically craft misleading input efficiently and without being noticed

AI security essentials



Many experts and organizations contributed to this overview of essentials - including close collaboration with SANS Institute, ensuring alignment with SANS' Critical AI security guidelines. SANS and the AI Exchange have an ongoing collaboration to share expertise and support broad education.

The upcoming sections provide overviews of AI security threats and controls.

Threats overview

> **Category: discussion**
> **Permalink: https://owaspai.org/goto/threatsoverview/**

Scope of Threats

In the AI Exchange we focus on AI-specific threats, meaning threats to AI assets (see **#SEC PROGRAM**, such as model parameters. Threats to other assets are already covered in many other resources - for example the protection of a user database. AI systems are IT systems so they suffer from various security threats. Therefore, when securing AI systems, the AI Exchange needs to be seen as an extension of your existing security program: AI security = threats to AI-specific assets (AI Exchange) +threats to other assets (other resources)

Threat Model

We distinguish between three types of threats:

1. threats during development-time (when data is obtained and prepared, and the model is

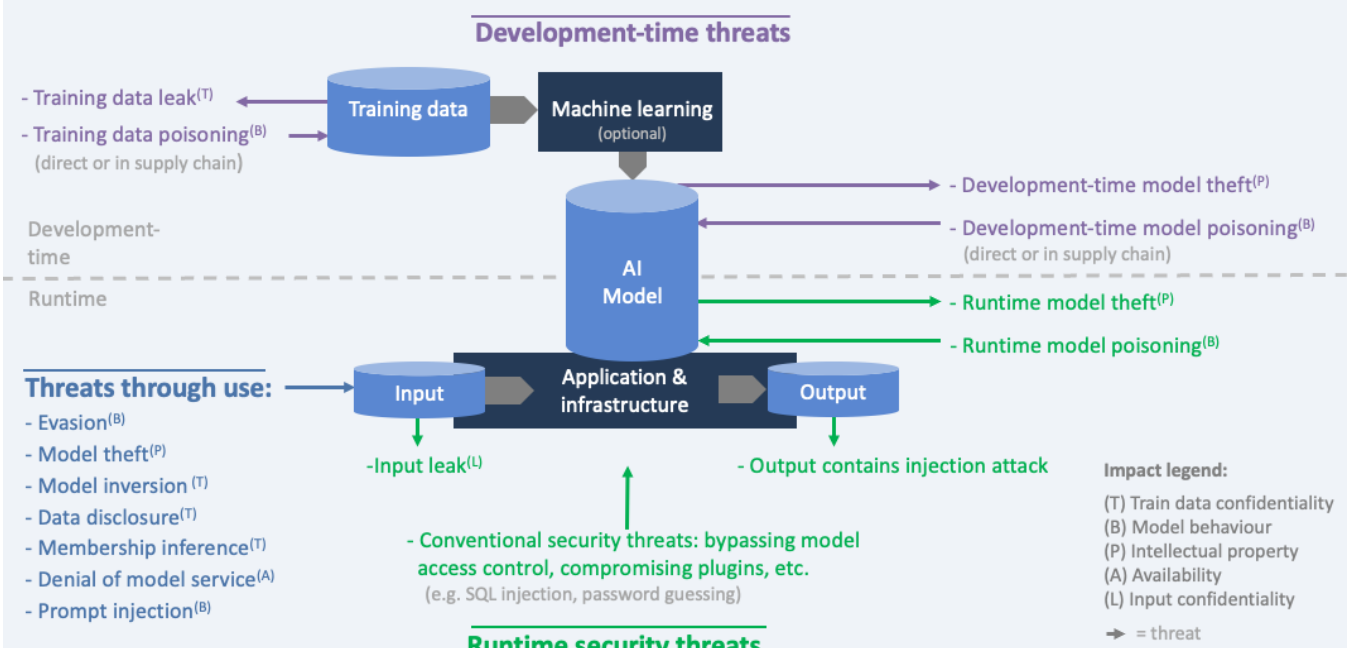
1. threats during development-time (when data is obtained and prepared, and the model is trained/obtained) - for example data poisoning
2. threats through using the model (through inference; providing input and getting the output) - for example prompt injection or evasion
3. other threats to the system during runtime (in operation - not through inference) - for example stealing model input

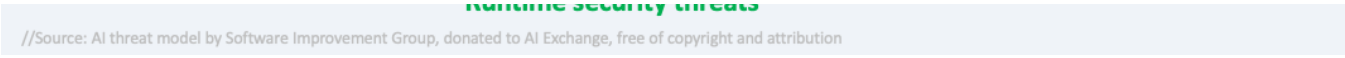
In AI, we outline 6 types of impacts that align with three types of attacker goals (disclose, deceive and disrupt):

1. disclose: hurt confidentiality of train/test data
2. disclose: hurt confidentiality of model Intellectual property (the *model parameters* or the process and data that led to them)
3. disclose: hurt confidentiality of input data
4. deceive: hurt integrity of model behaviour (the model is manipulated to behave in an unwanted way and consequentially, deceive users)
5. disrupt: hurt availability of the model (the model either doesn't work or behaves in an unwanted way - not to deceive users but to disrupt normal operations)
6. disrupt/disclose: confidentiality, integrity, and availability of non AI-specific assets

The threats that create these impacts use different attack surfaces. For example: the confidentiality of training data can be compromised by hacking into the database during development, but it can also get leaked by a *membership inference attack* that can find out whether a certain individual was in the train data, simply by feeding that person's data into the model and looking at the details of the model output.

The diagram shows the threats as arrows. Each threat has a specific impact, indicated by letters referring to the Impact legend. The control overview section contains this diagram with groups of controls added.





Note that some threats represent attacks consisting of several steps, and therefore present multiple threats in one, for example: — An adversary performs a data poisoning attack by hacking into the training database and placing poisoned samples, and then after the data has been used for training, presents specific inputs to make use of the corrupted behaviour. — An adversary breaks into a development environment to steal a model so it can be used to experiment on to craft manipulated inputs to achieve a certain goal, and then present that input to the deployed system.

Threats to agentic AI

> **Category: discussion**
> **Permalink: https://owaspai.org/goto/agenticaithreats/**

In Agentic AI, AI systems can take action instead of just present output, and sometimes act autonomously or communicate with other agents. Important note: these are still software systems and AI systems, so everything in the AI Exchange applies, but there are a few attention points.

An example of Agentic AI is a set of voice assistants that can control your heating, send emails, and even invite more assistants into the conversation. That's powerful—but you'd probably want it to check with you first before sending a thousand emails.

There are four typical properties of agentic AI:

1. Action: Agents don't just chat — they invoke functions such as sending an email. That makes **LEAST MODEL PRIVILEGE** a key control.
2. Autonomous: Agents can trigger each other, enabling autonomous responses (e.g., a script receives an email, triggering a GenAI follow-up). That makes **OVERSIGHT** important, and it makes working memory an attack vector because that's where the state and the plan of an autonomous agent lives.
3. Complex: Agentic behaviour is emergent.
4. Multi-system: You often work with a mix of systems and interfaces. Because of that, developers tend to assign responsibilities regarding access control to the AI using instructions, opening up the door for manipulation through **prompt injection**.

What does this mean for security?

- Hallucinations and prompt injections can change commands — or even escalate privileges. Key controls are defense in depth and blast radius control (**impact limitation**). Don't assign the responsibility of access control to GenAI models/agents. Build that into your architecture.

- Existing assumptions about things like trust boundaries and other established security measures might need to be revisited because agentic AI changes interconnectivity and data flows between system components.
- Agents deployed with their own sets of permissions open up privilege escalation vectors because they are susceptible to becoming a confused deputy
- The attack surface is wide, and the potential impact should not be underestimated.
- Because of that, the known controls become even more important — such as security of inter-model communication (e.g., MCP), **traceability**, protecting memory integrity, **prompt injection defenses**, **rule-based / human oversight**, and **least model privilege**. See the **controls overview section**.

For leaking sensitive data in agentic AI, you need three things, also called the *lethal trifecta*:

1. Data: Control of the attacker of data that find its way into an LLM at some point in the session of a user that has the desired access, to perform **indirect prompt injection**
2. Access: Access of that LLM or connected agents to sensitive data
3. Send: The ability of that LLM or connected agents to initiate sending out data to the attacker

See **Simon Willison's excellent work** for more details, and for examples in agentic AI software development **here** and **here**.

Prompt injection and mostly the **indirect** form is the key threat in most agentic AI systems. See the **seven layers section** on how these controls form layers of protection. After model alignment and filtering and detection, it should be assumed that prompt injection can still happen and therefore it is critical that *blast radius control* is performed.

Further links:

- For more details on the agentic AI threats, see the **Agentic AI threats and mitigations, from the GenAI security project**.
- For a more general discussion of Agentic AI, see **this article from Chip Huyen**.
- The **testing section** discusses more about agentic AI red teaming and links to the collaboration between CSA and the Exchange: the Agentic AI red teaming guide.
- **OWASP Agentic AI security top 10** plus **Rock's blog on it**

AI Security Matrix

> **Category: discussion**

> Permalink: https://owaspai.org/goto/aisecuritymatrix/

The AI security matrix below (click to enlarge) shows the key threats and risks, ordered by type and impact.

AI-specific?	Lifecycle	Attack surface	Threat/Risk category	Asset	Impacted	Unwanted result
AI	Operation	Model use (provide input/ read output)	Direct prompt injection	Model behaviour	Integrity	Manipulated unwanted model behaviour causes wrong decisions leading to business financial loss, misbehaviour going undetected, reputational damage, legal and compliance issues, operational disruption, customer dissatisfaction and churn, reduced empoloyee morale, incorrect strategic decisions, liabiity issues, personal damage and safety issues
			Indirect prompt injection			
			Evasion (e.g. adversarial examples)			
		Break into deployed model	Model poisoning in runtime (reprogramming)			
	Development	Engineering environment	Model poisoning development time			
			Data poisoning of train/finetune data			
		Supply chain	Model poisoning in supply chain (transfer learning attack)			
			Data poisoning in supply chain			
	Operation	Model use	Data disclosure in model output	Training data	Confidentiality	Leaking sensitive data can cause costs from fines and legal fees and remediation effort, loss of business through customer churn, reputation damage, loss of competitive advantage in case of trade secrets, operational disruption, impacted business relationships, and employee morale issues
			Model inversion / Membership inference			
	Development	Engineering environment	Training data leak			
	Operation	Model use	Model theft through use (input-output harvesting)	Model intellectual property	Confidentiality	If attackers can copy a model, the investment in the model is devalued caused by loss of competitive advantage, plus a copy can help craft (evasion) attacks
		Break into deployed model	Runtime model theft (not through use)			
	Development	Engineering environment	Model theft development-time			
	Operation	Model use	Denial of model service (model resource depletion)	Model behaviour	Availability	The model is not available, leading to business continuity issues, or safety problems
	Operation	All IT	Model input leak	Model input data	Confidentiality	Sensitive data in model input leaks. E.g. an LLM prompt with a sensitive question, enhanced with retrieved company secrets
	Operation	All IT	Model output contains injection attack	Any asset	C, I, A	Injection attack (from model output) causes harm
Generic	Operation	All IT	Generic runtime security attack	Any asset	C, I, A	Generic runtime security attack causes harm (includes social engineering/phishing)
	Development	All IT	Generic supply chain attack	Any asset	C, I, A	Generic supply chain security attack causes harm (e.g. vulnerability in a component)

Clickable version, based on the **Periodic table**:

Asset & Impact	Attack surface with lifecycle	Threat/Risk category
Model behaviour Integrity	Runtime -Model use (provide input/ read output)	**Direct prompt injection**
		Indirect prompt injection
		Evasion (e.g., adversarial examples)
	Runtime - Break into deployed model	**Model poisoning runtime** (reprogramming)
	Development -Engineering environment	**Development-environment model poisoning**
		Data poisoning of train/ finetune data
	Development - Supply chain	**Supply-chain model poisoning**

Training data Confidentiality	Runtime - Model use	**Sensitive data output from model**
		Model inversion / Membership inference
	Development - Engineering environment	**Developmen-time data leak**
Model confidentiality	Runtime - Model use	**Model theft through use** (input-output harvesting)
	Runtime - Break into deployed model	**Direct model theft runtime**
	Development - Engineering environment	**Model theft development-time**
Model behaviour Availability	Model use	**AI resource exhaustion**
Model input data Confidentialiy	Runtime - All IT	**Leak sensitive input data**
Any asset, CIA	Runtime-All IT	**Model output contains injection**
Any asset, CIA	Runtime - All IT	Conventional runtime security attack on conventional asset
Any asset, CIA	Runtime - All IT	Conventional attack on conventional supply chain

Controls overview

> Category: discussion
> Permalink: https://owaspai.org/goto/controlsoverview/

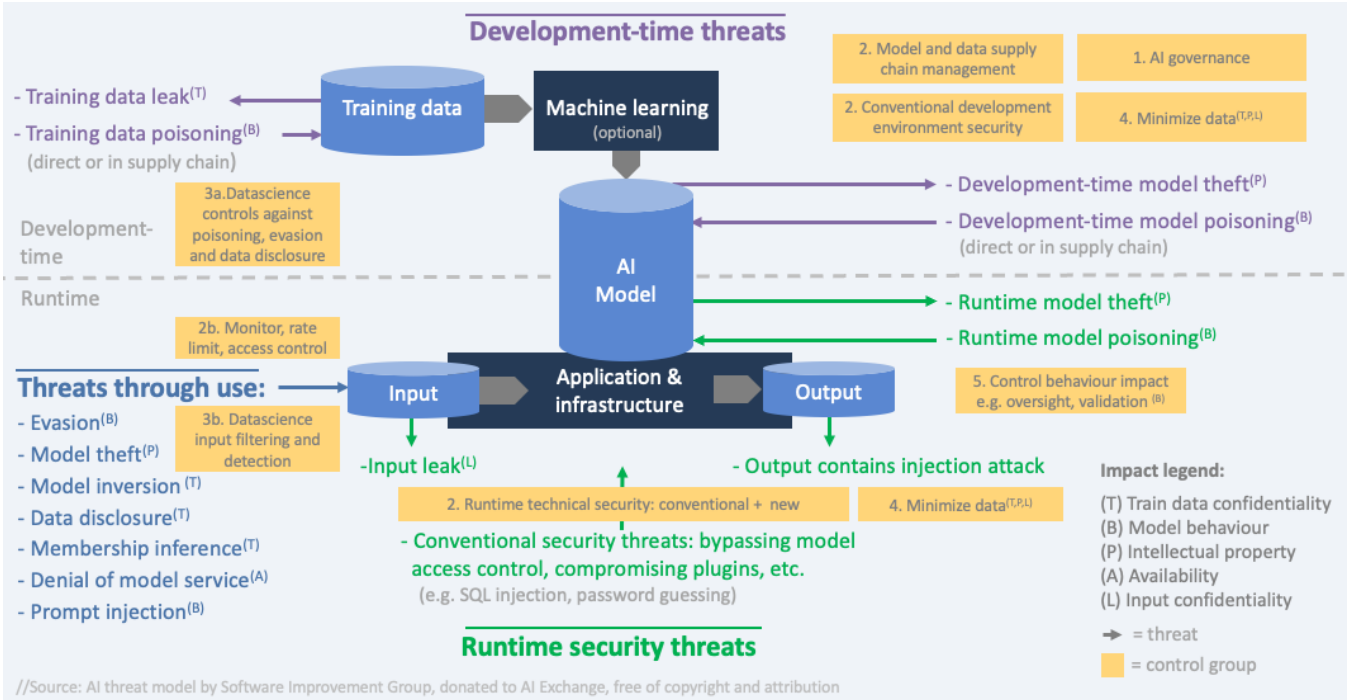
Select and implement controls with care

The AI exchange lists a number of controls to mitigate risks of attack. Be aware that many of the controls are expensive to implement and are subject to trade-offs with other AI properties that can affect accuracy and normal operations of the model. Particularly, controls that involve changes to the learning process and data distributions can have un-intented downstream side-effects, and must be considered and introduced with care.

Scope of controls In the AI Exchange we focus on AI-specific threats and their corresponding controls. Some of the controls are AI-specific (e.g., adding noise to the training set) and others are not (e.g., encrypting the training database). We refer to the latter as 'conventional controls'. The Exchange focuses on the details of the AI-specific controls because the details of conventional controls are specified elsewhere - see for example **OpenCRE**. We do provide AI-specific aspects of those controls, for example that protection of model parameters can be implemented using a Trusted Execution Environment.

Threat model with controls - general

The below diagram puts the controls in the AI Exchange into groups and places these groups in the right lifecycle with the corresponding threats.



The groups of controls form a summary of how to address AI security (controls are in capitals):

- **AI Governance**(1): integrate AI comprehensively into your information security and software development lifecycle processes, not just by addressing AI risks, but by embedding AI considerations across the entire lifecycle:

  > AI PROGRAM, SEC PROGRAM, DEV PROGRAM, SECDEV PROGRAM, CHECK COMPLIANCE, SEC EDUCATE

- **Extend supply chain management**(2) with data and model governance:

  > SUPPLY CHAIN MANAGE

> SUPPLY CHAIN MANAGE

- **Minimize/obfuscate data:**(4) Limit the amount of data at rest and in transit. Also, limit data storage time, development-time and runtime:

  > (DATA MINIMIZE, ALLOWED DATA, SHORT RETAIN, OBFUSCATE TRAINING DATA)

- Apply conventional **technical security controls**(2), since an AI system is an IT system:
  - Apply standard **conventional security controls** (e.g., 15408, ASVS, OpenCRE, ISO 27001 Annex A, NIST SP800-53) to the complete AI system and don't forget the new AI-specific assets :
    - Development-time: model & data storage, model & data supply chain, data science documentation:

      > DEV SECURITY, SEGREGATE DATA, DISCRETE

    - Runtime: model storage, model use, plug-ins, and model input/output:

      > RUNTIME MODEL INTEGRITY, RUNTIME MODEL IO INTEGRITY, RUNTIME MODEL CONFIDENTIALITY, MODEL INPUT CONFIDENTIALITY, ENCODE MODEL OUTPUT, LIMIT RESOURCES

  - **Adapt** conventional IT security controls to make them more suitable for AI (e.g., which usage patterns to monitor for):

    > MONITOR USE, MODEL ACCESS CONTROL, RATE LIMIT

  - Adopt **new** IT security controls:

    > CONF COMPUTE, MODEL OBFUSCATION, INPUT SEGREGATION

- Apply **AI engineer security controls**(3) :
  - Data/model engineering controls(3a) as part of development:

    > FEDERATED LEARNING, CONTINUOUS VALIDATION, UNWANTED BIAS TESTING, EVASION ROBUST MODEL, POISON ROBUST MODEL, TRAIN ADVERSARIAL, TRAIN DATA DISTORTION, ADVERSARIAL ROBUST DISTILLATION, MODEL ENSEMBLE, MORE TRAINDATA, SMALL MODEL, DATA QUALITY CONTROL, MODEL ALIGNMENT

  - Model I/O handling(3b) during runtime to filter and detect attacks:

    > ANOMALOUS INPUT HANDLING, EVASION INPUT HANDLING, UNWANTED INPUT SERIES HANDLING, PROMPT INJECTION I/O HANDLING, DOS INPUT VALIDATION, INPUT DISTORTION, SENSITIVE OUTPUT HANDLING, OBSCURE CONFIDENCE

- **Limit model behaviour**(5) as the model can behave in unwanted ways - unintentionally or by manipulation:

> OVERSIGHT, LEAST MODEL PRIVILEGE, MODEL ALIGNMENT, AI TRANSPARENCY, EXPLAINABILITY, CONTINUOUS VALIDATION, UNWANTED BIAS TESTING

All threats and controls are explored in more detail in the corresponding threat sections of the AI Exchange.

Threat model with controls - ready-made model

> **Category: discussion**
> **Permalink: https://owaspai.org/goto/readymademodel/**

If possible, and depending on price, organisations can prefer to use a ready-made model, instead of training or fine-tuning themselves. For example: an open source model to detect people in a camera image, or a general purpose LLM such as Google Gemini, OpenAI ChatGPT, Anthropic Claude, Alibaba QWen, Deepseek, Mistral, Grok or Falkon. Training such models yourself can cost millions of dollars, requires deep expertise and vast amounts of data.

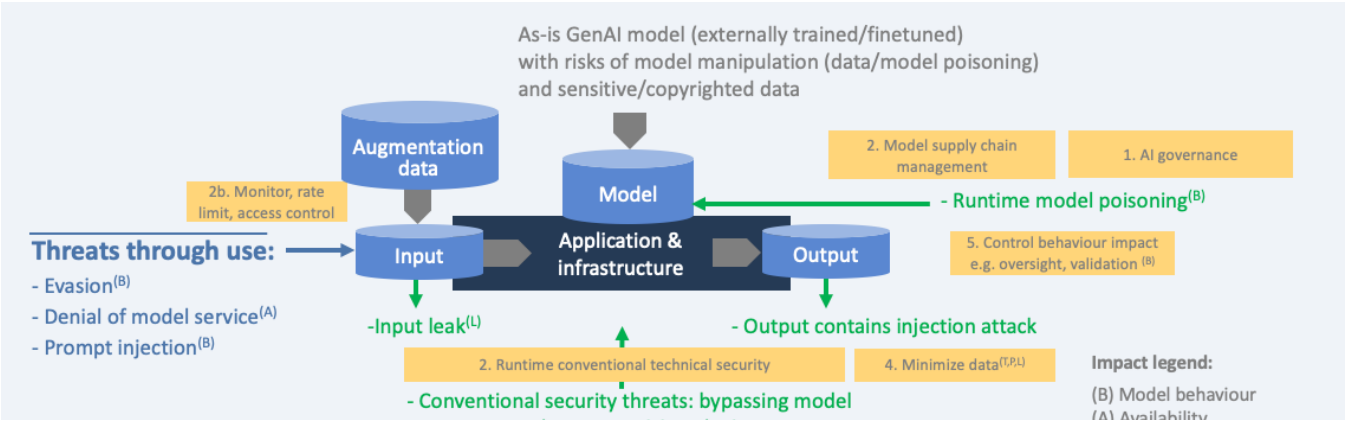
The provider (e.g., OpenAI) has done the training/fine tuning and therefore is responsible for part of security. Hence, proper supply chain management regarding the model provider is required.

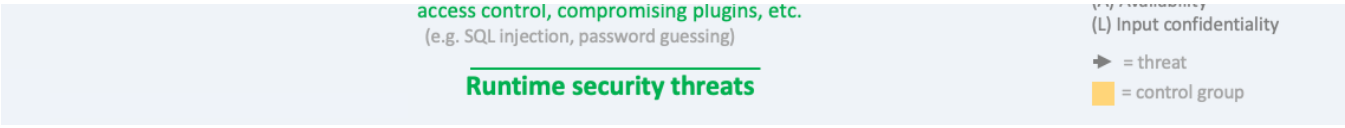
The following deployment options apply for ready-made models:

- Closed source model, hosted by the provider - for the largest models typically the only available option
- Self-hosted: Open source model (open weights) deployed on-premise (most secure) or in the virtual private cloud (secure if the cloud provider is trusted) - these options provide more security and may be the best option cost-wise, but do not support the largest models
- Open source model (open weights) at a paid hosting service - convenient

Self-hosted

The diagram below shows threats and controls in a self-hosting situation.





External-hosted

If the model is hosted externally, security largely depends on how the supplier handles data, including the security configuration. Some relevant questions to ask here include:

1. Where does the model run?
   Is the model running in the vendor's processes or in your own virtual private cloud? Some vendors say you get a 'private instance', but that may refer to the API, and not the model. If the model runs on the cluster operated by your vendor, your data leaves your environment in clear text. Vendors will minimize storage and transfer, but they may log and monitor.

2. What are the data retention rules?
   Has a court required the vendor to retain logs for litigation? This happened to OpenAI in the US for a period of time.

3. What is exactly logged and monitored?
   Read the small print. Is logging enabled, and if so, what is logged? And what is monitored - by operators or by algorithms? And in the case of monitoring algorithms: how is that infrastructure protected? Some vendors allow you to opt out of logging, but only with specific licenses.

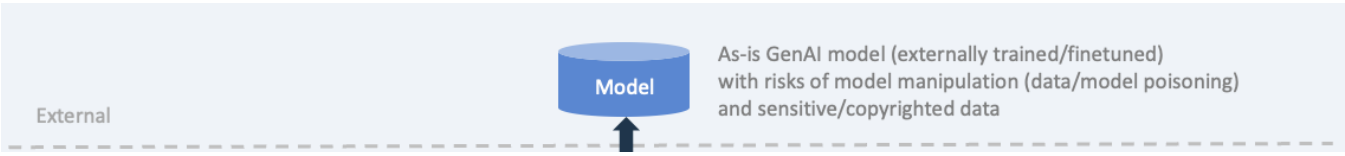
4. Is your input used for training?
   This is a common fear, but in the vast majority of cases the input is not used. If vendors would do this secretly, it would get out because there are ways to tell.

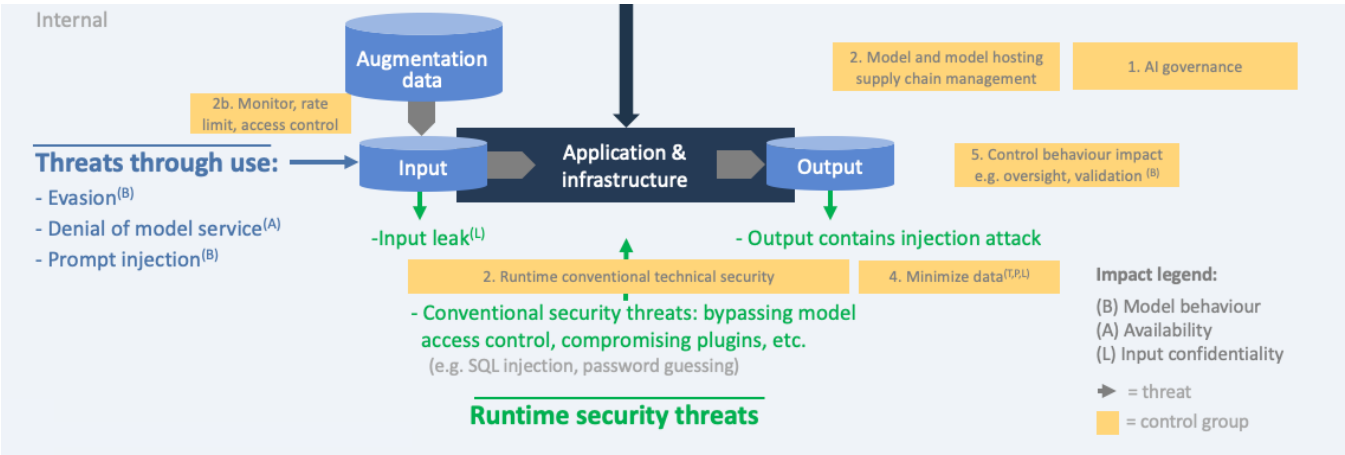
If you can't accept the risk for certain data, then hosting your own (smaller) model is the safest option. Typically it won't be as good and there's the catch 22.

It is important to realise that a provider-hosted model needs your input data in clear text, because the model must read the data to process it. This means your sensitive data will exist unencrypted outside your infrastructure.
This is not unique to LLM providers — it is the same for other multi-tenant SaaS services, such as commercial hosted Office suites. Even though providers usually minimise data storage, limit retention, and reduce data movement, the fact remains: your data leaves your environment in readable form.

When weighing this risk, compare it fairly: the vendor may still protect that environment better than you can protect your own.

The diagram below shows threats and controls in an externally hosted situation.





A typical challenge for organizations is to control the use of ready-made-models for general purpose Generative AI (e.g., ChatGPT), since employees typically can access many of them, even for free. Some of these models may not satisfy the organization's requirements for security and privacy. Still, employees can be very tempted to use them with the lack of a better alterative, sometimes referred to as *shadow AI*. The best solution for this problem is to provide a good alternative in the form of an AI model that has been deployed and configured in a secure and privacy-preserving way, of sufficient quality, and complying with the organization's values and policies. In addition, the risks of shadow AI need to be made very clear to users.

Periodic table of AI security

> **Category: discussion**
> **Permalink: 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 deep links to comprehensive coverage here at the AI Exchange website.
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, Monitor, rate limit, model access control plus: Prompt injection I/O handling, Model alignment
		Indirect prompt injection	Limit unwanted behavior, Prompt

		injection	behavior, Prompt injection I/O handling, Input segregation
		Evasion (e.g., adversarial examples)	Limit unwanted behavior, Monitor, rate limit, model access control plus: Anomalous input handling, Evasion input handling, Unwanted input series handling, 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	Development-environment model poisoning	Limit unwanted behavior, Development environment security, data segregation, federated learning, supply chain management plus: model ensemble
		Data poisoning of train/finetune data	Limit unwanted behavior

		train/finetune data	behavior, Development environment security, data segregation, federated learning, supply chain management plus: model ensemble plus: More training data, data quality control, train data distortion, poison robust model, train adversarial
	Development - Supply chain	Supply-chain model poisoning	Limit unwanted behavior, Supplier: Development environment security, data segregation, federated learning Producer: supply chain management plus: model ensemble
Training data Confidentiality	Runtime - Model use	Data disclosure in model output	Sensitive data limitation (data minimize, short retain, obfuscate training data) plus: Monitor, rate limit, model access control plus:

			Sensitive output handling
		Model inversion / Membership inference	**Sensitive data limitation** (data minimize, short retain, obfuscate training data) plus: **Monitor**, **rate limit**, **model access control** plus: Unwanted input series handling, >**Obscure confidence**, **Small model**
	Development - Engineering environment	**Training data leaks**	**Sensitive data limitation** (data minimize, short retain, obfuscate training data) plus: **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** plus: **Unwanted input series handling**
	Runtime - Break into deployed model	**Direct model theft runtime**	**Runtime model confidentiality**, **Model obfuscation**

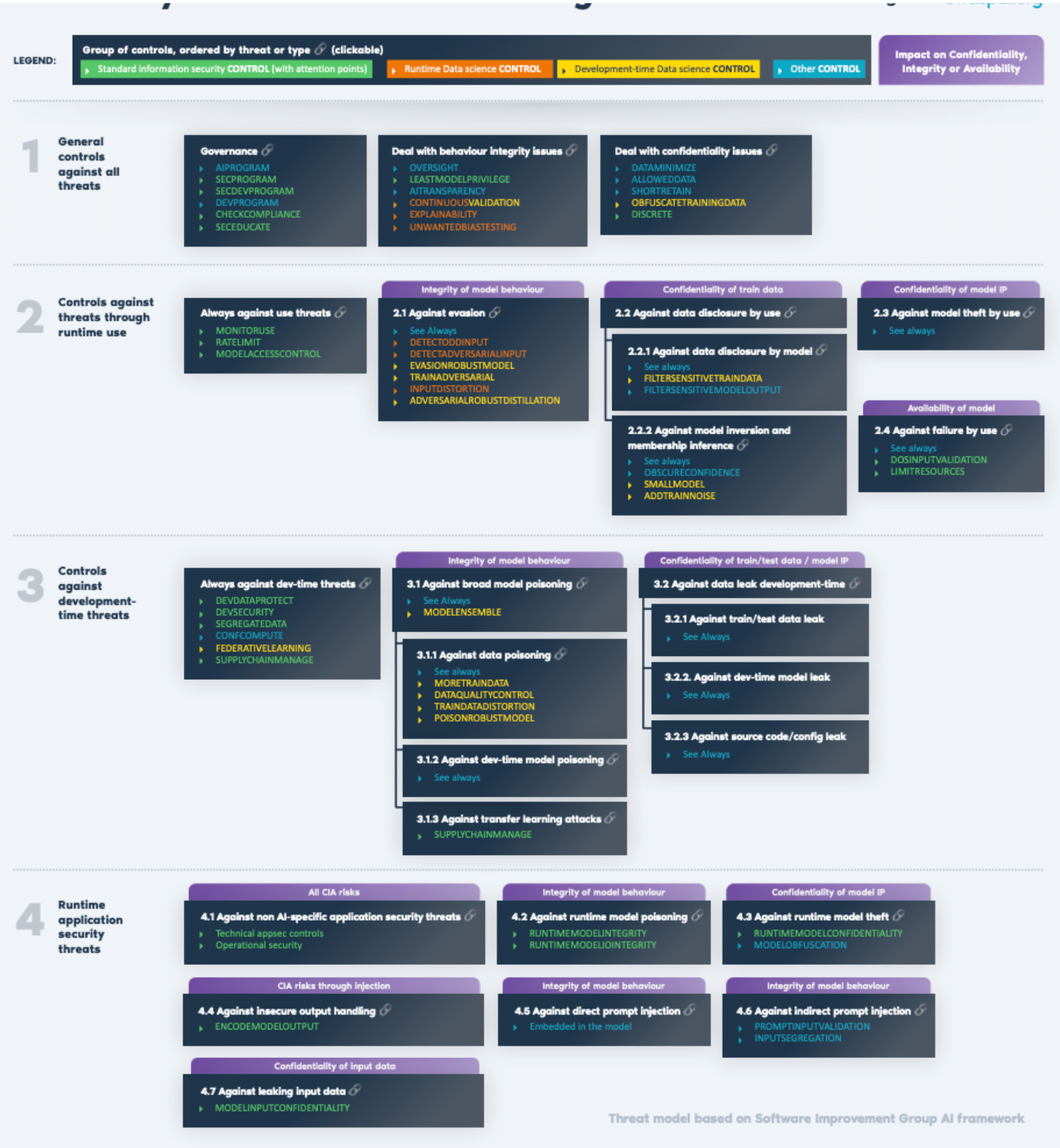
	Development - Engineering environment	**Model theft development-time**	**Development environment security**, **data segregation**, **federated learning**
Model behaviour Availability	Model use	**AI resource exhaustion** (model resource depletion)	**Monitor**, **rate limit**, **model access control** plus: **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

Structure of threats and controls in the deep dive section

> **Category: discussion**
> **Permalink:** https://owaspai.org/goto/navigator/

The next big section in this document is an extensive deep dive into all the AI security threats and their controls.
The navigator diagram below outlines the structure of the deep-dive section, illustrating the relationships between threats, controls, associated risks, and the types of controls applied.

ℹ️ Click on the image to get a PDF with clickable links.



How to select relevant threats and controls? risk analysis

> Category: discussion
> Permalink: https://owaspai.org/goto/riskanalysis/

There are quite a number of threats and controls described in this document. The relevance and severity of each threat and the appropriate controls depend on your specific use case and how AI is deployed within your environment. Determining which threats apply, to what extent, and who is responsible for implementing

your environment. Determining which threats apply, to what extent, and who is responsible for implementing controls should be guided by a risk assessment based on your architecture and intended use. Simply go to the 'Identifying risks' section below and follow the steps.

Risk management introduction

Organizations classify their risks into several key areas: Strategic, Operational, Financial, Compliance, Reputation, Technology, Environmental, Social, and Governance (ESG). A threat becomes a risk when it exploits one or more vulnerabilities. AI threats, as discussed in this resource, can have significant impact across multiple risk domains. For example, adversarial attacks on AI systems can lead to disruptions in operations, distort financial models, and result in compliance issues. See the **AI security matrix** for an overview of AI related threats, risks and potential impact.

General risk management for AI systems is typically driven by AI governance - see **AIPROGRAM** and includes both risks BY relevant AI systems and risks to those systems. Security risk assessment is typically driven by the security management system - see **SECPROGRAM** as this system is tasked to include AI assets, AI threats, and AI systems provided that these have been added to the corresponding repositories. ISO/IEC 27005 is the international standard for security risk management.

Organizations often adopt a Risk Management framework, commonly based on ISO 31000 or similar standards such as ISO 23894. These frameworks guide the process of managing risks through four key steps as outlined below:

1. **Identifying Risks**: Recognizing potential risks that could impact the organization or others.
2. **Evaluating Risks**:
   By estimating the likelihood and severity of the impact should the risk materialize, it is necessary to assess the probability of the risk occurring and evaluating the potential consequences should the risk materialize. The likelihood and severity combined represent the level of the risk. This is typically presented in the form of a heatmap with combinations of likelihood versus severity.
3. **Risk Treatment**: Risk treatment means choosing an appropriate strategy to address the risk. These strategies include: Risk Mitigation, Transfer, Avoidance, or Acceptance. See below for further details.
4. **Risk Communication and Monitoring**:
   Regularly sharing risk information with stakeholders to ensure awareness and continuous support for risk management activities. Ensuring effective Risk Treatments are applied. This requires a Risk Register, a comprehensive list of risks and their attributes (e.g., severity, treatment plan, ownership, status, etc.). This is discussed in more detail in the sections that follow.
5. Repeat the above process regularly and when changes warrant it.

Let's go through the risk management steps one by one.

1. Identifying Risks - decision tree

Discovering potential risks that could impact the organization requires the technical and business assessment of the applicable threats. In this document, we focus on AI-specific risks only - meaning risks to the AI-specific assets. The following section takes you through each type of risk impact, to identify the risks that apply in your case.

Identify risks of unwanted model behaviour

Regarding model behaviour, we focus on manipulation by attackers, as the scope of this document is security. Other sources of unwanted behavior are general inaccuracy (e.g., hallucinations) and/or unwanted bias regarding certain groups (discrimination).

This will always be an applicable threat, independent of your use-case, simply because the model behaviour matters by definition. Nevertheless, the risk level may sometimes be accepted as shown below.

This means that you always need to have in place the following:

- **General governance controls** (e.g., maintaining a documented inventory of AI applications and implementing mechanisms to ensure appropriate oversight and accountability.)
- **Controls to limit effects of unwanted model behaviour** (e.g., human oversight when necessary, model least privilege for agents)

Question: Is the model GenAI (e.g., a Large Language Model)?

- Protect against **prompt injection** in case a) an attacker can provide input to the model (e.g., a prompt), and b) the model could theoretically create output that results in harm - for example: offensive output, dangerous information, misinformation, or triggering harmful functions (Agentic AI). The first question is: has the model supplier done enough according to your risk appetite. For this, you can check tests that the supplier or others have performed tests and when not available: do these tests yourself. What you accept, in other words: what you find too much effort in combination with too harmful, depends on your context. If a user wants the AI to say something offensive: do you regard it as a problem if that user succeeds in getting offended? Do you regard it as a problem if users can get a recipe to make poison - given that they can get this from many other AI's out there. See the linked threat section for more details.
- Protect against **indirect prompt injection** when your system inserts untrusted data in a prompt e.g. you retrieve somebody's resume and include it in a prompt, or an agentic system retrieves data that is untrusted.

Question: Who trains/finetunes the model?

- The supplier: protect against **Supply chain model poisoning**: obtaining or working with a model that has been manipulated to behave in unintended ways. This is done through proper **supply chain management** (e.g., selecting a trustworthy supplier and verifying the authenticity of the model). This is to gain assurance on the security posture of the provider, meaning the provider prevents model poisoning during development, including data poisoning, and uses uncompromised data. If the risk of data poisoning remains unacceptable, implementing post-training countermeasures can be an option if you have the expertise and if you have access to the model parameters (e.g., open source weights). See **POISONROBUSTMODEL**. Note that providers are typically not very open about their security countermeasures, which means that it can be challenging to gain sufficient assurance. Regulations will hopefully help achieve more provider transparency. For more details, see **ready made models**.
- You: you need to protect against **development-time model poisoning** which includes model poisoning, data poisoning and obtaining poisoned data or a poisoned pre-trained model in case you're finetuning the model.

Why not train/finetune a model yourself? There are many third party and open source models that may be able to perform the required task, perhaps after some fine tuning. Organizations often choose external GenAI models because they are typically general purpose, and training is difficult and expensive (often millions of dollars). Finetuning of generative AI is also not often performed by organizations given the cost of compute and the complexity involved. Some GenAI models can be obtained and run on your own infrastructure. The reasons for this can be lower cost (if it is an open source model), and the fact that sensitive input information does not have to be sent externally. A reason to use an externally hosted GenAI model can be the quality of the model.

Question: Do you use RAG (Retrieval Augmented Generation) ? Yes: Then your retrieval repository plays a role in determining the model behaviour. This means:

- You need to protect against **leaking** or **manipulation** of your augmentation data (e.g., vector database), which includes preventing that it contains externally obtained poisoned data.

Question: Who runs the model?

- The supplier: select a trustworthy supplier through **supply chain management**, to make sure the deployed model cannot be manipulated (**runtime model poisoning**) - just the way you would expect any supplier to protect their running application from manipulation.
- You: You need to protect against **runtime model poisoning** where attackers change the

model that you have deployed.

Question: Is the model (predictive AI or Generative AI) used in a classification task (e.g., spam or fraud detection)?

- Yes: Protect against an **evasion attack** in which a user tries to fool the model into a wrong decision using data (not instructions). Here, the level of risk is an important aspect to evaluate - see below. The risk of an evasion attack may be acceptable.

In order to assess the level of risk for unwanted model behaviour through manipulation, consider what the motivation of an attacker could be. What could an attacker gain by for example sabotaging your model? Just a claim to fame? Could it be a disgruntled employee? Maybe a competitor? What could an attacker gain by a less conspicuous model behaviour attack, like an evasion attack or data poisoning with a trigger? Is there a scenario where an attacker benefits from fooling the model? An example where evasion IS interesting and possible: adding certain words in a spam email so that it is not recognized as such. An example where evasion is not interesting is when a patient gets a skin disease diagnosis based on a picture of the skin. The patient has no interest in a wrong decision, and also the patient typically has no control - well maybe by painting the skin. There are situations in which this CAN be of interest for the patient, for example to be eligible for compensation in case the (faked) skin disease was caused by certain restaurant food. This demonstrates that it all depends on the context whether a theoretical threat is a real threat or not. Depending on the probability and impact of the threats, and on the relevant policies, some threats may be accepted as risk. When not accepted, the level of risk is input to the strength of the controls. For example: if data poisoning can lead to substantial benefit for a group of attackers, then the training data needs to be given a high level of protection.

Identify risks of leaking training data

Question: Do you train/finetune the model yourself?

- If yes, is the training data sensitive? If so, you need to protect against:
  - **unwanted disclosure in model output**
  - **model inversion**
  - **training data leaking from your engineering environment**.
  - **membership inference** - but only when the fact that something or someone was part of the training data constitutes sensitive information. For example, when the training set consists of criminals and their history to predict criminal careers. Membership of that set gives away the person is a convicted or alleged criminal.

Question: do you use RAG?

- Yes: apply the above to your augmentation data, as if it was part of the training set: as the repository data feeds into the model and can therefore be part of the output as well.

If you don't train/finetune the model, then the supplier of the model is responsible for unwanted content in the training data. This can be poisoned data (see above), data that is confidential, or data that is copyrighted. It is important to check licenses, warranties and contracts for these matters, or accept the risk based on your circumstances.

Identify risks of model theft

Question: Do you train/finetune the model yourself?

- If yes, is the model regarded as intellectual property? Then you need to protect against:
  - **Model theft through use**
  - **Model theft development-time**
  - **Source code/configuration leak**
  - **Runtime model theft**

Identify risks of leaking input data

Question: Is your input data sensitive?

- Protect against **leaking input data**. Especially if the model is run by a supplier, proper care needs to be taken to ensure that this data is minimized and transferred or stored securely. Review the security measures provided by the supplier, including any options to disable logging or monitoring on their end. Realise that most Cloud AI models have your input and output unencrypted in their infrastructure (just like documents in Google Suite and Microsoft 365). If you use the right license and configuration, you can prevent it from being stored or analysed. One risk that remains is that the government of the supplier may be forced to store and keep input and output to serve for subpoenas. If you're using a RAG system, remember that the data you retrieve and inject into the prompt also counts as input data. This often includes sensitive company information or personal data.

Identify further risks

Question: Does your model create text output?

- Protect against **insecure output handling**, for example, when you display the output of the model on a website and the output contains malicious Javascript.

Make sure to protect against **model unavailability by malicious users** (e.g., large inputs, many requests). If your model is run by a supplier, then certain countermeasures may already be in place to address this.

Since AI systems are software systems, they require appropriate conventional application security and operational security, apart from the AI-specific threats and controls mentioned in this section

operational security, apart from the AI-specific threats and controls mentioned in this section.

2. Evaluating Risks by Estimating Likelihood and Impact

To determine the severity of a risk, it is necessary to assess the likelihood of the risk occurring and evaluating the potential consequences should the risk materialize.

Estimating the Likelihood:
Estimating the likelihood and impact of an AI risk requires a thorough understanding of both the technical and contextual aspects of the AI system in scope. The likelihood of a risk occurring in an AI system is influenced by several factors, including the complexity of the AI algorithms, the data quality and sources, the conventional security measures in place, and the potential for adversarial attacks. For instance, an AI system that processes public data is more susceptible to data poisoning and inference attacks, thereby increasing the likelihood of such risks. A financial institution's AI system, which assesses loan applications using public credit scores, is exposed to data poisoning attacks. These attacks could manipulate creditworthiness assessments, leading to incorrect loan decisions.

Examples of aspects involved in rating probability:

- Opportunity regarding attacker access (OWASP, FAIR - Factor Analysis for Information Risk)
- Risk of getting caught (FAIR)
- Capabilities/tools/budget (ISO/IEC 27005, OWASP, FAIR)
- Susceptibility of the system (ISO/IEC 27005, FAIR)
- Motive(OWASP, FAIR, ISO/IEC 27005)
- Number of potential attackers(OWASP)
- Data regarding incidents and attempts (ISO/IEC 27005)

Evaluating the Impact: Evaluating the impact of risks in AI systems involves understanding the potential consequences of threats materializing. This includes both the direct consequences, such as compromised data integrity or system downtime, and the indirect consequences, such as reputational damage or regulatory penalties. The impact is often magnified in AI systems due to their scale and the critical nature of the tasks they perform. For instance, a successful attack on an AI system used in healthcare diagnostics could lead to misdiagnosis, affecting patient health and leading to significant legal, trust, and reputational repercussions for the involved entities.

Prioritizing risks The combination of likelihood and impact assessments forms the basis for prioritizing risks and informs the development of Risk Treatment decisions. Commonly, organizations use a risk heat map to visually categorize risks by impact and likelihood. This approach facilitates risk communication and decision-making. It allows the management to focus on risks with highest severity (high likelihood and high impact).

3. Risk Treatment

Risk treatment is about deciding what to do with the risks: transfer, avoid, accept, or mitigate. Mitigation involves selecting and implementing controls. This process is critical due to the unique vulnerabilities and threats related to AI systems such as data poisoning, model theft, and adversarial attacks. Effective risk treatment is essential to robust, reliable, and trustworthy AI.

Risk Treatment options are:

1. **Mitigation**: Implementing controls to reduce the likelihood or impact of a risk. This is often the most common approach for managing AI cybersecurity risks. See the many controls in this resource and the 'Select controls' subsection below.
   - Example: Enhancing data validation processes to prevent data poisoning attacks, where malicious data is fed into the Model to corrupt its learning process and negatively impact its performance.
2. **Transfer**: Shifting the risk to a third party, typically through transfer learning, federated learning, insurance or outsourcing certain functions. - Example: Using third-party cloud services with robust security measures for AI model training, hosting, and data storage, transferring the risk of data breaches and infrastructure attacks.
3. **Avoidance**: Changing plans or strategies to eliminate the risk altogether. This may involve not using AI in areas where the risk is deemed too high. - Example: Deciding against deploying an AI system for processing highly sensitive personal data where the risk of data breaches cannot be adequately mitigated.
4. **Acceptance**: Acknowledging the risk and deciding to bear the potential loss without taking specific actions to mitigate it. This option is chosen when the cost of treating the risk outweighs the potential impact. - Example: Accepting the minimal risk of model inversion attacks (where an attacker attempts to reconstruct publicly available input data from model outputs) in non-sensitive applications where the impact is considered low.

4. Risk Communication & Monitoring

Regularly sharing risk information with stakeholders to ensure awareness and support for risk management activities.

A central tool in this process is the Risk Register, which serves as a comprehensive repository of all identified risks, their attributes (such as severity, treatment plan, ownership, and status), and the controls implemented to mitigate them. Most large organizations already have such a Risk Register. It is important to align AI risks and chosen vocabularies from Enterprise Risk Management to facilitate effective communication of risks throughout the organization.

5. Arrange responsibility

For each selected threat, determine who is responsible for addressing it. By default, the organization that builds and deploys the AI system is responsible, but building and deploying may be done by different organizations, and some parts of the building and deployment may be deferred to other organizations, e.g. hosting the model, or providing a cloud environment for the application to run. Some aspects are shared responsibilities.

If some components of your AI system are hosted, then you share responsibility regarding all controls for the relevant threats with the hosting provider. This needs to be arranged with the provider by using a tool like the responsibility matrix. Components can be the model, model extensions, your application, or your infrastructure. See **Threat model of using a model as-is**.

If an external party is not open about how certain risks are mitigated, consider requesting this information and when this remains unclear you are faced with either 1) accept the risk, 2) or provide your own mitigations, or 3) avoid the risk, by not engaging with the third party.

6. Verify external responsibilities

For the threats that are the responsibility of other organisations: attain assurance whether these organisations take care of it. This would involve the controls that are linked to these threats.

Example: Regular audits and assessments of third-party security measures.

7. Select controls

Next, for the threats that are relevant to your use-case and fall under your responsibility, review the associated controls, both those listed directly under the threat (or its parent category) and the general controls, which apply universally. See the **Periodic table** for an overview of which controls mitigate the risks for each threat. For each control, consider its purpose and assess whether it's worth implementing, and to what extent. This decision should weigh the cost of implementation against how effectively the control addresses the threat, along with the level of the associated risk. These factors also influence the order in which you apply controls. Start with the highest-risk threats and prioritize low-cost, quick-win controls (the "low-hanging fruit").

Controls often have quality-related parameters that need to be adjusted to suit the specific situation and level of risk. For example, this could involve deciding how much noise to add to input data or setting appropriate thresholds for anomaly detection. Testing the effectiveness of these controls in a simulation environment helps you evaluate their performance and security impact to find the right balance. This tuning process should be continuous, using insights from both simulated tests and real-world production feedback.

When have you done enough? The AI system is sufficiently secure when all identified risks can be treated, meaning transferred, avoided or accepted, where acceptance in some cases can be done directly, without

meaning transferred, avoided or accepted, where acceptance in some cases can be done directly, without first taking action, and in other cases require you to implement controls to bring the risk to an acceptable level.

8. Residual risk acceptance

In the end, you need to be able to accept the risks that remain regarding each threat, given the controls that you implemented.

9. Further management of the selected controls

(see **SECPROGRAM**), which includes continuous monitoring, documentation, reporting, and incident response.

10. Continuous risk assessment

Implement continuous monitoring to detect and respond to new threats. Update the risk management strategies based on evolving threats and feedback from incident response activities.
Example: Regularly reviewing and updating risk treatment plans to adapt to new vulnerabilities.

How about ...

How about AI outside of machine learning?

A helpful way to look at AI is to see it as consisting of machine learning (the current dominant type of AI) models and *heuristic models*. A model can be a machine learning model which has learned how to compute based on data, or it can be a heuristic model engineered based on human knowledge, e.g. a rule-based system. Heuristic models still require data for testing, and in some cases, for conducting analysis that supports further development and validation of human-derived knowledge.
This document focuses on machine learning. Nevertheless, here is a quick summary of the machine learning threats from this document that also apply to heuristic systems:

- Model evasion is also possible with heuristic models, as attackers may try to find loopholes or weaknesses in the defined rules.
- Model theft through use - it is possible to train a machine learning model based on input/output combinations from a heuristic model.
- Overreliance in use - heuristic systems can also be relied on too much. The applied knowledge can be false.
- Both data poisoning and model poisoning can occur by tampering with the data used to

- Both data poisoning and model poisoning can occur by tampering with the data used to enhance knowledge, or by manipulating the rules either during development or at runtime.
- Leaks of data used for analysis or testing can still be an issue.
- Knowledge base, source code and configuration can be regarded as sensitive data when it is intellectual property, so it needs protection.
- Leak sensitive input data, for example when a heuristic system needs to diagnose a patient.

How about responsible or trustworthy AI?

> **Category: discussion**
> **Permalink: https://owaspai.org/goto/responsibleai/**

There are many aspects of AI when it comes to positive outcomes while mitigating risks. This is often referred to as responsible AI or trustworthy AI, where the former emphasises ethics, society, and governance, while the latter emphasises the more technical and operational aspects.

If your primary responsibility is security, it's best to start by focusing on AI security. Once you have a solid grasp of that, you can expand your knowledge to other AI aspects, even if it's just to support colleagues who are responsible for those areas and help them stay vigilant. After all, security professionals are often skilled at spotting potential failure points. Furthermore, some aspects can be a consequence of compromised AI and are therefore helpful to understand, such as *safety*.

Let's break down the principles of AI and explore how each one connects to security:

- **Accuracy** is about the AI model being sufficiently correct to perform its 'business function'. Being incorrect can lead to harm, including (physical) safety problems (e.g., car trunk opens during driving) or other wrong decisions that are harmful (e.g., wrongfully declined loan). The link with security is that some attacks cause unwanted model behaviour which is by definition, an accuracy problem. Nevertheless, the security scope is restricted to mitigating the risks of those attacks - NOT solve the entire problem of creating an accurate model (selecting representative data for the trainset etc.).
- **Safety** refers to the condition of being protected from / unlikely to cause harm. Therefore safety of an AI system is about the level of accuracy when there is a risk of harm (typically implying physical harm but not restricted to that), plus the things that are in place to mitigate those risks (apart from accuracy), which includes security to safeguard accuracy, plus a number of safety measures that are important for the business function of the model. These need to be taken care of and not just for security reasons because the model can make unsafe decisions for other reasons (e.g., bad training data), so they are a shared concern between safety and security

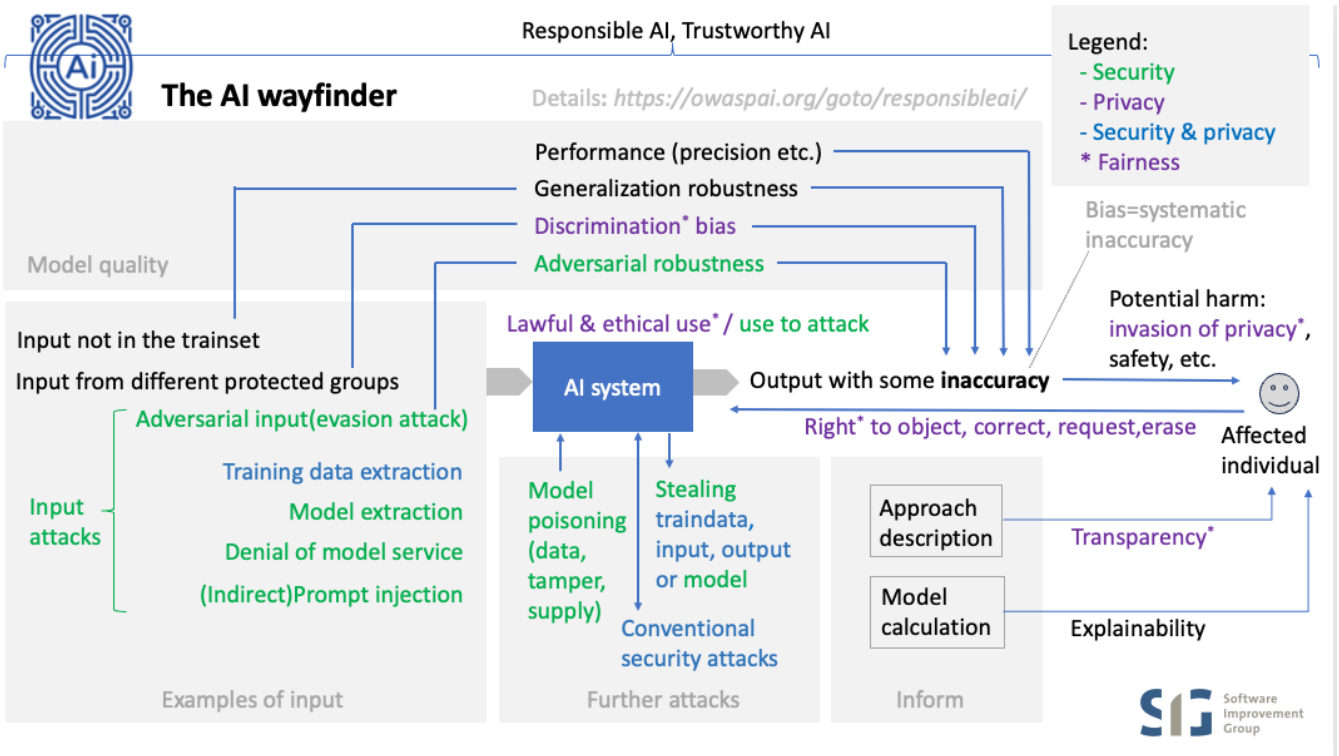
shared concern between safety and security:

- **oversight** to restrict unsafe behaviour, and connected to that: assigning least privileges to the model,
- **continuous validation** to safeguard accuracy,
- **transparency**: see below,
- **explainability**: see below.

- **Transparency**: sharing information about the approach, to warn users and depending systems of accuracy risks, plus in many cases users have the right to know details about a model being used and how it has been created. Therefore it is a shared concern between security, privacy and safety.
- **Explainability**: sharing information to help users validate accuracy by explaining in more detail how a specific result came to be. Apart from validating accuracy this can also support users to get transparency and to understand what needs to change to get a different outcome. Therefore it is a shared concern between security, privacy, safety and business function. A special case is when explainability is required by law separate from privacy, which adds 'compliance' to the list of aspects that share this concern.
- **Robustness** is about the ability of maintaining accuracy under expected or unexpected variations in input. The security scope is about when those variations are malicious (*adversarial robustness*) which often requires different countermeasures than those required against normal variations (*generalization robustness*). Just like with accuracy, security is not involved per se in creating a robust model for normal variations. The exception is when generalization robustness or adversarial robustness is involved, as this becomes a shared concern between safety and security. Whether it falls more under one or the other depends on the specific case.
- **Free of discrimination**: without unwanted bias of protected attributes, meaning: no systematic inaccuracy where the model 'mistreats' certain groups (e.g. gender, ethnicity). Discrimination is undesired for legal and ethical reasons. The relation with security is that having detection of unwanted bias can help to identify unwanted model behaviour caused by an attack. For example, a data poisoning attack has inserted malicious data samples in the training set, which at first goes unnoticed, but then is discovered by an unexplained detection of bias in the model. Sometimes the term 'fairness' is used to refer to discrimination issues, but mostly fairness in privacy is a broader term referring to fair treatment of individuals, including transparency, ethical use, and privacy rights.
- **Empathy**. Its connection to security lies in recognizing the practical limits of what security can achieve when evaluating an AI application. If individuals or organizations cannot be

adequately protected, empathy means rethinking the idea, either by rejecting it altogether or by taking additional precautions to reduce potential harm.

- **Accountability**. The relation of accountability with security is that security measures should be demonstrable, including the processes that have led to those measures. In addition, traceability as a security property is important, just like in any IT system, in order to detect, reconstruct and respond to security incidents and provide accountability.
- **AI security**. The security aspect of AI is the central topic of the AI Exchange. In short, it can be broken down into:
  - **Input attacks**, that are performed by providing input to the model
  - **Model poisoning**, aimed to alter the model's behavior
  - Stealing AI assets, such as train data, model input, output, or the model itself, either **development time** or runtime (see below)
  - Further **runtime conventional security attacks**



How about Generative AI (e.g. LLM)?

> Category: discussion
> Permalink: https://owaspai.org/goto/genai/

Yes, GenAI is leading the current AI revolution and it's the fastest moving subfield of AI security. Nevertheless it is important to realize that other types of algorithms (let's call it *predictive AI*) will remain to be applied to many important use cases such as credit scoring, fraud detection, medical diagnosis, product

recommendation, image recognition, predictive maintenance, process control, etc. Relevant content has been marked with 'GenAI' in this document.

Important note: from a security threat perspective, GenAI is not that different from other forms of AI (*predictive AI*). GenAI threats and controls largely overlap and are very similar to AI in general. Nevertheless, some risks are (much) higher. Some are lower. Only a few risks are GenAI-specific. Some of the control categories differ substantially between GenAI and predictive AI - mostly the data science controls (e.g. adding noise to the training set). In many cases, GenAI solutions will use a model as-is and not involve any training by the organization whatsoever, shifting some of the security responsibilities from the organization to the supplier. Nevertheless, if you use a **ready-made model**, you need still to be aware of those threats.

What is mainly new to the threat landscape because of LLMs?

- First of all, LLMs pose new threats to security because they may be used to create code with vulnerabilities, or they may be used by attackers to create malware, or they may cause harm through hallucinations. However, these concerns are outside the scope of the AI Exchange, which focuses on security threats to AI systems themselves.
- Regarding input:
  - Prompt injection is when attackers manipulate the behaviour of the model with crafted and sometimes hidden instructions.
  - Also new is organizations sending huge amounts of data in prompts, with company secrets and personal data.
- Regarding output: The fact that output can contain injection attacks, or can contain sensitive or copyrighted data is new (see **Copyright**).
- Overreliance is an issue. We let LLMs control and create things and may have too much trust in how correct they are, and also underestimate the risk of them being manipulated. The result is that attacks can have much impact.
- Regarding training: Since the training sets are so large and based on public data, it is easier to perform data poisoning. Poisoned foundation models are also a big supply chain issue.
- Just like any AI system, a generative AI system can trigger actions based on the output, but in the case of generative AI, the model output can contain function calls to perform actions (e.g. send mail) or trigger other AI systems. See **Agentic AI** for more details.

GenAI security particularities are:

Nr.	GenAI security particularities	OWASP for LLM TOP 10

1	GenAI models are controlled by natural language in prompts, creating the risk of **Prompt injection**. Direct prompt injection is where the user tries to fool the model to behave in unwanted ways (e.g. offensive language), whereas with indirect prompt injection it is a third party that injects content into the prompt for this purpose (e.g. manipulating a decision).	(**OWASP for LLM 01:Prompt injection**)
2	GenAI models have typically been trained on very large datasets, which makes it more likely to output **sensitive data** or **licensed data**, for which there is no control of access privileges built into the model. All data will be accessible to the model users. Some mechanisms may be in place in terms of system prompts, model alignment, or output filtering, but those are typically not watertight.	(**OWASP for LLM 02: Sensitive Information Disclosure**)
3	**Data and model poisoning** is an AI-broad problem, and with GenAI the risk is generally higher since training data can be supplied from different sources that may be challenging to control, such as the internet. Attackers could for example hijack domains and place manipulated information.	(**OWASP for LLM 04: Data and Model Poisoning**)

4	GenAI models can be inaccurate and hallucinate. This is an AI-broad risk factor, and Large Language Models (GenAI) can make matters worse by coming across as very confident and knowledgeable. In essence, this is about the risk of underestimating the probability that the model is wrong or the model has been manipulated. This means that it is connected to each and every security control. The strongest link is with **controls that limit the impact of unwanted model behavior**, in particular **Least model privilege**.	(**OWASP for LLM 06: Excessive agency**) and (**OWASP for LLM 09: Misinformation**)
5	**Leaking input data**: GenAI models mostly live in the cloud - often managed by an external party, which increases the risk of leaking prompts. This issue is not limited to GenAI, but GenAI has 2 particular risks here: 1) model use involves user interaction through prompts, adding user data and corresponding privacy/ sensitivity issues, and 2) GenAI model input (prompts) can contain rich context information with sensitive data (e.g. company secrets). The latter issue occurs with *in-context learning* or *Retrieval Augmented Generation (RAG)* (adding	Not covered in LLM top 10

	background information to a prompt): for example data from all reports ever written at a consultancy firm. First of all, this information will travel with the prompt to the cloud, and second: the system will likely not respect the original access rights to the information.	
6	Pre-trained models may have been manipulated. The concept of pretraining is not limited to GenAI, but the approach is quite common in GenAI, which increases the risk of **supply-chain model poisoning**.	(**OWASP for LLM 03 - Supply chain vulnerabilities**)
7	**Model inversion and membership inference** are typically low to zero risks for GenAI.	Not covered in LLM top 10, apart from LLM06 which uses a different approach - see above
8	GenAI output may contain elements that perform an **injection attack** such as cross-site-scripting.	(**OWASP for LLM 05: Improper Output Handling**)
9	**Resource exhaustion** can be an issue for any IT system, but GenAI models typically cost more to run, so overloading them can create unwanted costs.	(**OWASP for LLM 10: Unbounded consumption**)

GenAI References:

- **OWASP LLM top 10**
- **Demystifying the LLM top 10**

- **Impacts and risks of GenAI**
- **LLMsecurity.net**

How about the NCSC/CISA guidelines?

> **Category: discussion**
> **Permalink: https://owaspai.org/goto/jointguidelines/**

Mapping of the UK NCSC /CISA **Joint Guidelines for secure AI system development** to the controls here at the AI Exchange.
To see those controls linked to threats, refer to the **Periodic table of AI security**.

Note that the UK Government drove an initiative through their DSIT department to build on these joint guidelines and produce the **DSIT Code of Practice for the Cyber Security of AI**, which reorganizes things according to 13 principles, does a few tweaks, and adds a bit more of governance. The principle mapping is added below, and adds mostly post-market aspects:

- Principle 10: Communication and processes associated with end-users and affected entities
- Principle 13: Ensure proper data and model disposal

1. Secure design

- Raise staff awareness of threats and risks (DSIT principle 1):
  #SECURITY EDUCATE
- Model the threats to your system (DSIT principle 3):
  See Risk analysis under **#SECURITY PROGRAM**
- Design your system for security as well as functionality and performance (DSIT principle 2):
  #AI PROGRAM, **#SECURITY PROGRAM**, **#DEVELOPMENT PROGRAM**, **#SECURE DEVELOPMENT PROGRAM**, **#CHECK COMPLIANCE**, **#LEAST MODEL PRIVILEGE**, **#DISCRETE**, **#OBSCURE CONFIDENCE**, **#OVERSIGHT**, **#RATE LIMIT**, **#DOS INPUT VALIDATION**, **#LIMIT RESOURCES**, **#MODEL ACCESS CONTROL**, **#AI TRANSPARENCY**
- Consider security benefits and trade-offs when selecting your AI model
  All development-time data science controls (currently 13), **#EXPLAINABILITY**

2. Secure Development

- Secure your supply chain (DSIT principle 7):
  #SUPPLY CHAIN MANAGE

- Identify, track and protect your assets (DSIT principle 5):
  #DEVELOPMENT SECURITY, #SEGREGATE DATA, #CONFIDENTIAL COMPUTE, #MODEL INPUT CONFIDENTIALITY, #RUNTIME MODEL CONFIDENTIALITY, #DATA MINIMIZE, #ALLOWED DATA, #SHORT RETAIN, #OBFUSCATE TRAINING DATA and part of #SECURITY PROGRAM
- Document your data, models and prompts (DSIT principle 8):
  Part of #DEVELOPMENT PROGRAM
- Manage your technical debt:
  Part of #DEVELOPMENT PROGRAM

3. Secure deployment

- Secure your infrastructure (DSIT principle 6):
  Part of #SECURITY PROGRAM and see 'Identify, track and protect your assets'
- Protect your model continuously:
  #INPUT DISTORTION, #FILTER SENSITIVE MODEL OUTPUT, #RUNTIME MODEL IO INTEGRITY, #MODEL INPUT CONFIDENTIALITY, #PROMPT INJECTION I/O HANDLING, #INPUT SEGREGATION
- Develop incident management procedures:
  Part of #SECURITY PROGRAM
- Release AI responsibly:
  Part of #DEVELOPMENT PROGRAM
- Make it easy for users to do the right things (DSIT principle 4, called Enable human responsibility for AI systems):
  Part of #SECURITY PROGRAM, and also involving #EXPLAINABILITY, documenting prohibited use cases, and #HUMAN OVERSIGHT)

4. Secure operation and maintenance

- Monitor your system's behaviour (DSIT principle 12 and similar to DSIT principle 9 - appropriate testing and validation):
  #CONTINUOUS VALIDATION, #UNWANTED BIAS TESTING
- Monitor your system's inputs:
  #MONITOR USE, #DETECT ODD INPUT, #DETECT ADVERSARIAL INPUT
- Follow a secure by design approach to updates (DSIT principle 11: Maintain regular security updates, patches and mitigations):
  Part of #SECURE DEVELOPMENT PROGRAM

Part of #SECURE DEVELOPMENT PROGRAM

- Collect and share lessons learned:
  Part of #SECURITY PROGRAM and #SECURE DEVELOPMENT PROGRAM

How about copyright?

> **Category: discussion**
> **Permalink: https://owaspai.org/goto/copyright/**

Introduction

AI and copyright are two (of many) areas of law and policy, (both public and private), that raise complex and often unresolved questions. AI output or generated content is not yet protected by US copyright laws. Many other jurisdictions have yet to announce any formal status as to intellectual property protections for such materials. On the other hand, the human contributor who provides the input content, text, training data, etc. may own a copyright for such materials. Finally, the usage of certain copyrighted materials in AI training may be considered **fair use**.

AI & Copyright Security

In AI, companies face a myriad of security threats that could have far-reaching implications for intellectual property rights, particularly copyrights. As AI systems, including large data training models, become more sophisticated, they inadvertently raise the specter of copyright infringement. This is due in part to the need for development and training of AI models that process vast amounts of data, which may contain copyright works. In these instances, if copyright works were inserted into the training data without the permission of the owner, and without consent of the AI model operator or provider, such a breach could pose significant financial and reputational risk of infringement of such copyright and corrupt the entire data set itself.

The legal challenges surrounding AI are multifaceted. On one hand, there is the question of whether the use of copyrighted works to train AI models constitutes infringement, potentially exposing developers to legal claims. On the other hand, the majority of the industry grapples with the ownership of AI-generated works and the use of unlicensed content in training data. This legal ambiguity affects all stakeholders including developers, content creators, and copyright owners alike.

Lawsuits Related to AI & Copyright

Recent lawsuits (writing is April 2024) highlight the urgency of these issues. For instance, a class action suit filed against Stability AI, Midjourney, and DeviantArt alleges infringement on the rights of millions of artists by training their tools on web-scraped images.
Similarly, Getty Images' lawsuit against Stability AI for using images from its catalog without permission to train an art-generating AI underscores the potential for copyright disputes to escalate. Imagine the same scenario where a supplier provides vast quantities of training data for your systems that have been

scenario where a supplier provides vast quantities of training data for your systems that have been compromised by protected work, data sets, or blocks of materials not licensed or authorized for such use.

Copyright of AI-generated source code

Source code constitutes a significant intellectual property (IP) asset of a software development company, as it embodies the innovation and creativity of its developers. Therefore, source code is subject to IP protection, through copyrights, patents, and trade secrets. In most cases, human generated source code carries copyright status as soon as it is produced.

However, the emergence of AI systems capable of generating source code without human input poses new challenges for the IP regime. For instance, who is the author of the AI-generated source code? Who can claim the IP rights over it? How can AI-generated source code be licensed and exploited by third parties?

These questions are not easily resolved, as the current IP legal and regulatory framework does not adequately address the IP status of AI- generated works. Furthermore, the AI-generated source code may not be entirely novel, as it may be derived from existing code or data sources. Therefore, it is essential to conduct a thorough analysis of the origin and the process of the AI-generated source code, to determine its IP implications and ensure the safeguarding of the company's IP assets. Legal professionals specializing in the field of IP and technology should be consulted during the process.

As an example, a recent case still in adjudication shows the complexities of source code copyrights and licensing filed against GitHub, OpenAI, and Microsoft by creators of certain code they claim the three entities violated. More information is available here: **: GitHub Copilot copyright case narrowed but not neutered • The Register**

Copyright damages indemnification

Note that AI vendors have started to take responsibility for copyright issues of their models, under certain circumstances. Microsoft offers users the so-called **Copilot Copyright Commitment**, which indemnifies users from legal damages regarding copyright of code that Copilot has produced - provided **a number of things** including that the client has used content filters and other safety systems in Copilot and uses specific services. Google Cloud offers its **Generative AI indemnification**.
Read more at **The Verge on Microsoft indemnification** and **Direction Microsoft on the requirements of the indemnification**.

Do generative AI models really copy existing work?

Do generative AI models really look up existing work that may be copyrighted? In essence: no. A Generative AI model does not have sufficient capacity to store all the examples of code or pictures that were in its training set. Instead, during training, it extracts patterns about how things work in the data that it sees, and then later, based on those patterns, it generates new content. Parts of this content may show remnants of existing work, but that is more of a coincidence. In essence, a model doesn't recall exact blocks of code, but

uses its ‘understanding’ of coding to create new code. Just like with human beings, this understanding may result in reproducing parts of something you have seen before, but not per se because this was from exact memory. Having said that, this remains a difficult discussion that we also see in the music industry: did a musician come up with a chord sequence because she learned from many songs that this type of sequence works and then coincidentally created something that already existed, or did she copy it exactly from that existing song?

Mitigating Risk

Organizations have several key strategies to mitigate the risk of copyright infringement in their AI systems. Implementing them early can be much more cost effective than fixing at later stages of AI system operations. While each comes with certain financial and operating costs, the “hard savings” may result in a positive outcome. These may include:

1. Taking measures to mitigate the output of certain training data. The OWASP AI Exchange covers this through the corresponding threat: **data disclosure through model output**.
2. Comprehensive IP Audits: a thorough audit may be used to identify all intellectual property related to the AI system as a whole. This does not necessarily apply only to data sets but overall source code, systems, applications, interfaces and other tech stacks.
3. Clear Legal Framework and Policy: development and enforcement of legal policies and procedures for AI use, which ensure they align with current IP laws including copyright.
4. Ethics in Data Sourcing: source data ethically, ensuring all data used for training the AI models is either created in-house, or obtained with all necessary permissions, or is sourced from public domains which provide sufficient license for the organization’s intended use.
5. Define AI-Generated Content Ownership: clearly defined ownership of the content generated by AI systems, which should include under what conditions it can be used, shared, disseminated.
6. Confidentiality and Trade Secret Protocols: strict protocols will help protect confidentiality of the materials while preserving and maintaining trade secret status.
7. Training for Employees: training employees on the significance and importance of the organization’s AI IP policies along with implications on what IP infringement may be will help be more risk averse.
8. Compliance Monitoring Systems: an updated and properly utilized monitoring system will help check against potential infringements by the AI system.
9. Response Planning for IP Infringement: an active plan will help respond quickly and effectively to any potential infringement claims.
10. Additional mitigating factors to consider include seeking licenses and/or warranties from

AI suppliers regarding the organization's intended use, as well as all future uses by the AI system. With the help of a legal counsel, the organization should also consider other contractually binding obligations on suppliers to cover any potential claims of infringement.

Helpful resources regarding AI and copyright:

- Artificial Intelligence (AI) and Copyright | Copyright Alliance
- AI industry faces threat of copyright law in 2024 | Digital Watch Observatory
- Using generative AI and protecting against copyright issues | World Economic Forum -weforum.org
- Legal Challenges Against Generative AI: Key Takeaways | Bipartisan Policy Center
- Generative AI Has an Intellectual Property Problem - hbr.org
- Recent Trends in Generative Artificial Intelligence Litigation in the United States | HUB | K&L Gates - klgates.com
- Generative AI could face its biggest legal tests in 2024 | Popular Science - popsci.com
- Is AI Model Training Compliant With Data Privacy Laws? - termly.io
- The current legal cases against generative AI are just the beginning | TechCrunch
- (Un)fair Use? Copyrighted Works as AI Training Data — AI: The Washington Report | Mintz
- Potential Supreme Court clash looms over copyright issues in generative AI training data | VentureBeat
- AI-Related Lawsuits: How The Stable Diffusion Case Could Set a Legal Precedent | Fieldfisher

We are always happy to assist you!

Send us a message

Have questions about AI security? Want to contribute to our mission? We'd love to hear from you. Reach out through any of our channels or use the contact form.

First Name*

Last Name*

Email Address*

Message

Submit →

Advancing AI security through community collaboration and open-source resources.

Home

Overview

Media

Sponsor

Contribute

Connect

Designed & Developed by:

USABILITY DESIGNS