Chapter 1: AI and Cybersecurity

This book is about how to limit proliferation of bad AI in the world as this technology works it's way into every element of everyday living, from the internet to the IoT to transportation modes before it is too late to avert catastrophes caused by technology. With the rush to monetization we should not sacrifice security for a rush to ROI. However, that is not a realistic situation as shall be seen in this book that reviews current cybersecurity for AI and Humanoid Robots. Other elements that make cybersecurity a primary goal in developing AI is that it also makes AI more efficient and minimizes errors, so cybersecurity is value additive not subtractive to long term growth and sustainable monetization pathways. This works focus is on Large Language Models (LLMs), Large Reasoning Models (LRMs), AI Agents, and Humanoid Robots.

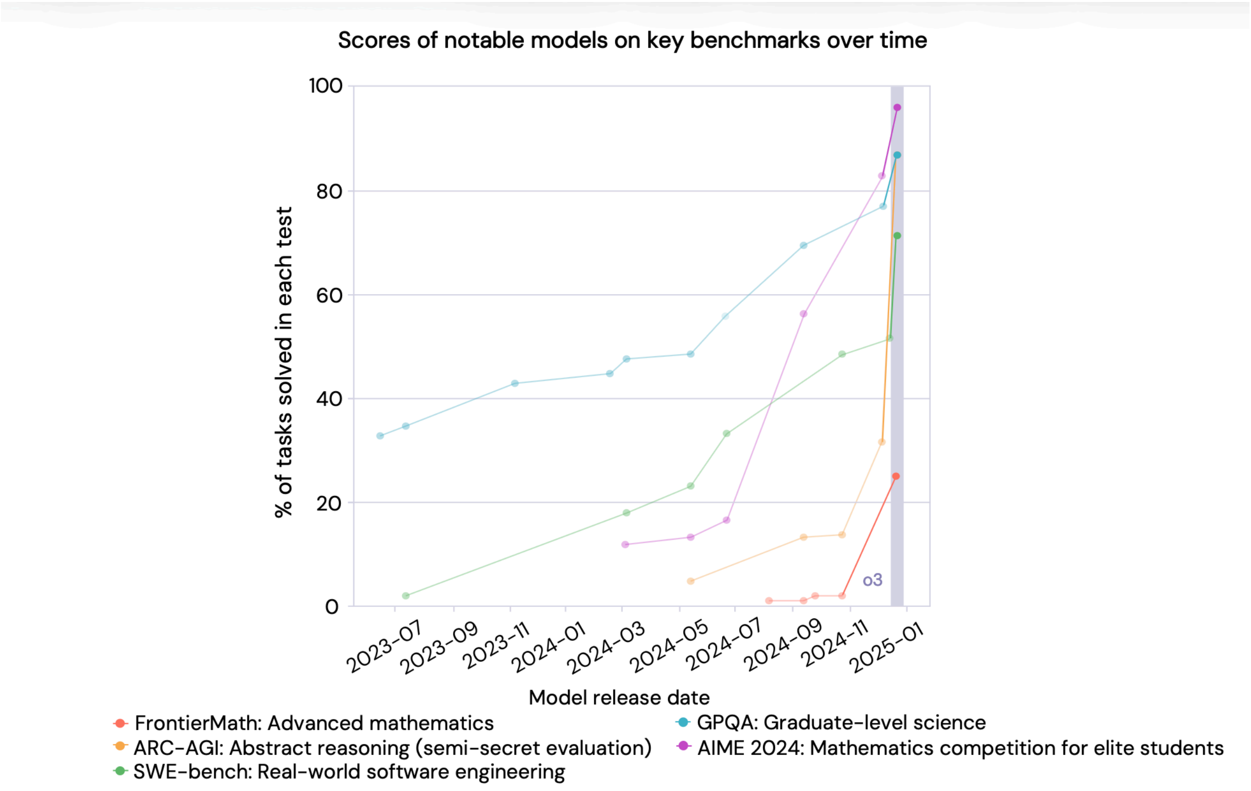


Figure 0.1: Scores of notable general-purpose AI models on key benchmarks from June 2023 to December 2024. o3 showed significantly improved performance compared to the previous state of the art (shaded region). These benchmarks are some of the field's most challenging tests of programming, abstract reasoning, and scientific reasoning. For the unreleased o3, the announcement date is shown; for the other models, the release date is shown. Some of the more recent AI models, including o3, benefited from improved scaffolding and more computation at test-time. Sources: Anthropic, 2024; Chollet, 2024; Chollet et al., 2025; Epoch AI, 2024; Glazer et al. 2024; OpenAI, 2024a; OpenAI, 2024b; Jimenez et al., 2024; Jimenez et al., 2025.

What AI Is, and Why Cybersecurity Matters

Artificial Intelligence (AI) has existed in various forms for decades, but the arrival of modern large language models (LLMs), multi-modal systems, and tool-using autonomous agents has transformed AI from a specialized research field into a pervasive, general-purpose capability. So what comprises a LLM? An LLM is created by following certain procedures, it all begins with acquiring data, then pre-training on that data, which is to say assigning maths in the form of weights to the underlying data being trained on. Once there is an underlying model it can be further refined based on purpose or function in the fine-tuning stage of the process. Some common definitions of the important components of the language model development cycle are:

- Model: A computer program, often based on machine learning, designed to process inputs and generate outputs. AI models can perform tasks such as prediction, classification, decision-making, or generation, forming the core of AI applications.
- Weights: Model parameters that represent the strength of connection between nodes in a neural network. Weights play an important part in determining the output of a model in response to a given input and are iteratively updated during model training to improve its performance.
- Fine-tuning: The process of adapting a pre-trained AI model to a specific task or making it more useful in general by training it on additional data.
- Pre-training: A stage in developing a general-purpose AI model in which models learn patterns from large amounts of data. The most compute-intensive stage of model development.
- System integration: The process of combining an AI model with other software components to produce a full ‘AI system’ that is ready for use. For instance, integration might consist in developers combining a general-purpose AI model with content filters and a user interface to produce a chatbot application.
- Data collection and pre-processing: A stage of AI development in which developers and data workers collect, clean, label, standardise, and transform raw training data into a format that the model can effectively learn from.

AI lifecycle: The distinct stages of developing AI, including data collection and pre-processing, pre-training, fine-tuning, model integration, deployment, post-deployment monitoring, and downstream modifications. Which is produced by the following process, that produces a generative AI such as the LLMs:

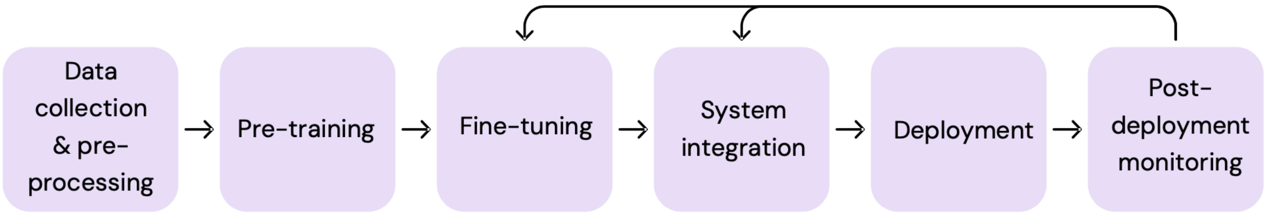
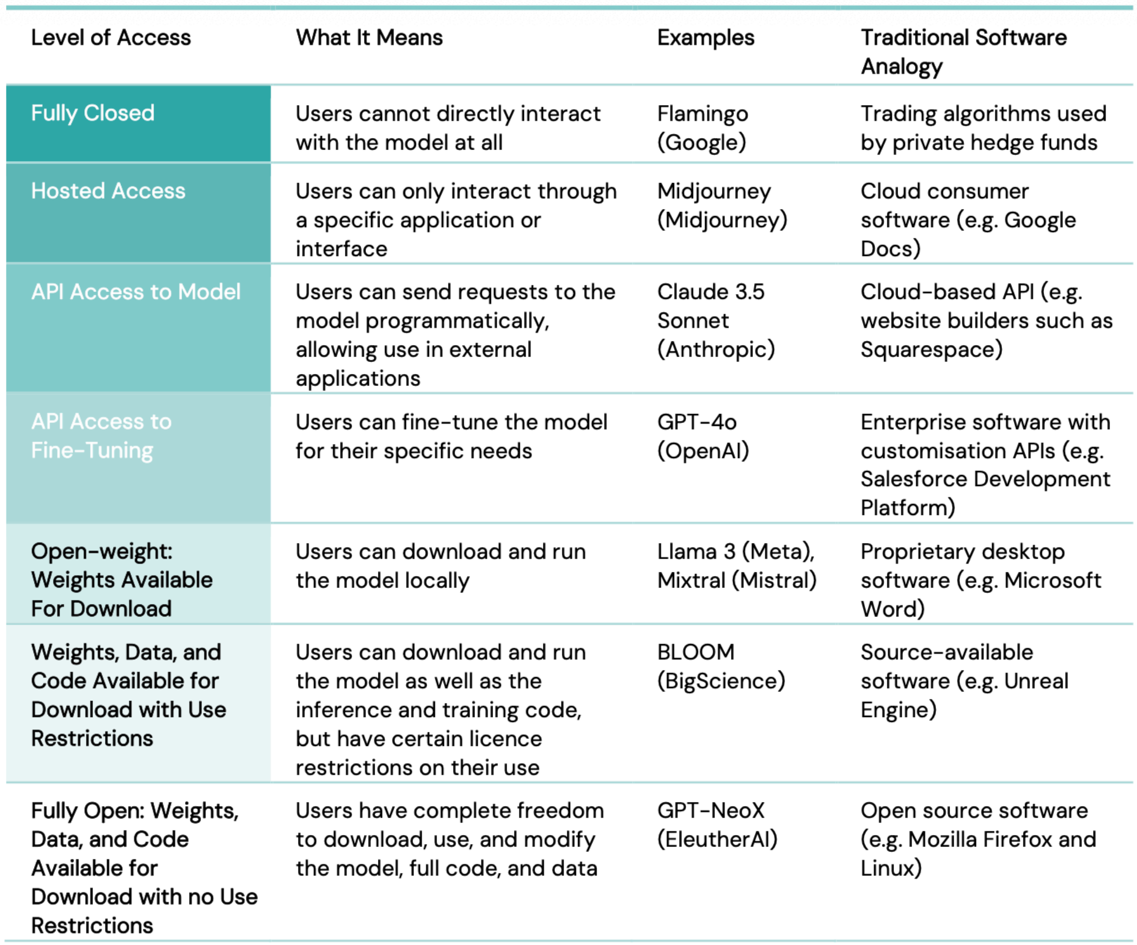


Figure 1.2: The process of developing and deploying general-purpose AI follows a series of distinct stages, from data collection and pre-processing to post-deployment monitoring. Source: International AI Safety Report.

Once the model such as ChatGPT, Claude, etc is derived there are enhancements and changes made in the post-deployment monitoring phase of the process which can then loop back to additional fine-tuning and model or system integration.

An LLM can be either closed source, proprietary, or open-source, which means others can use the model and modify it for their own purposes. There is also a spectrum of model release options from fully closed to fully open, all of which involve trade-offs between risks and benefits:

- Fully open models are open source models for which weights, full code, training data, and other documentation (e.g. about the model's training process) are made publicly available, without restrictions on modification, use and sharing. In general, fully open model release facilitates broader research and innovation but increases risks of malicious use by making it easy for malicious actors to bypass safety restrictions and modify the model for harmful purposes, and by increasing the likelihood of model flaws proliferating downstream into modified model versions and applications if downstream users do not proactively update the model version they use.



Level of Access	What It Means	Examples	Traditional Software Analogy
Fully Closed	Users cannot directly interact with the model at all	Flamingo (Google)	Trading algorithms used by private hedge funds
Hosted Access	Users can only interact through a specific application or interface	Midjourney (Midjourney)	Cloud consumer software (e.g. Google Docs)
API Access to Model	Users can send requests to the model programmatically, allowing use in external applications	Claude 3.5 Sonnet (Anthropic)	Cloud-based API (e.g. website builders such as Squarespace)
API Access to Fine-Tuning	Users can fine-tune the model for their specific needs	GPT-4o (OpenAI)	Enterprise software with customisation APIs (e.g. Salesforce Development Platform)
Open-weight: Weights Available For Download	Users can download and run the model locally	Llama 3 (Meta), Mixtral (Mistral)	Proprietary desktop software (e.g. Microsoft Word)
Weights, Data, and Code Available for Download with Use Restrictions	Users can download and run the model as well as the inference and training code, but have certain licence restrictions on their use	BLOOM (BigScience)	Source-available software (e.g. Unreal Engine)
Fully Open: Weights, Data, and Code Available for Download with no Use Restrictions	Users have complete freedom to download, use, and modify the model, full code, and data	GPT-NeoX (EleutherAI)	Open source software (e.g. Mozilla Firefox and Linux)

Table 2.5: There is a spectrum of model sharing options ranging from fully closed models (models are private and held only for proprietary use) to fully open, open source models (model weights, data, and code are freely and publicly available without restriction of use, modification, and sharing). This section focuses on the three rightmost columns.

- Fully

closed models' weights and code are proprietary, for internal use only. This means that external actors are not able to misuse the model and flaws are less likely to proliferate downstream and can be fixed once discovered. However, with closed models it is also harder for external developers to discover misuse risks, flaws, and use the model for wider innovation and research.

- Partially open models share some combination of weights, code, and data under various licences or access controls, in an attempt to balance the benefits of openness against risk mitigation and proprietary concerns. For example, OpenAI provides public access to its GPT-4o model through an interface called ChatGPT that allows users to prompt the system and retrieve responses without accessing the model itself. This kind of partial 'query access' allows the public to use the model and study its behaviour and performance flaws without providing direct access to the model weights and code. The cost of this partial access is that external AI researchers (academia and third-party evaluators) do not have access to perform deeper analysis of system safety, and downstream developers cannot freely integrate the model into new applications and products. Some licences such as RAIL (Responsible AI License) articulate restrictions against harmful uses of the model. Licence restrictions are legal articulations only and provide no physical barrier to misuse if the model itself is available for public download. Some actors may be deterred from misuse by the potential of legal liability, while other malicious actors may simply ignore the licence condition. (Bengio et al. 2025)

The question of open vs closed is directly relevant to the creation of secure and trustworthy agents, based on LLMs and LRMs. As malicious users such as cybercriminals can take an open source model and it's weights and put it to use to create hacking agents or influence agents for the purpose of financial gain, like in the phishing family of attacks.

LLM Guardrails

A central component in discussions of AI Cybersecurity for LLMs is that of the topic of Guardrails, which are intended as checks on then ability of the LLM to be used for nefarious purposes. OWASP relates:

> LLM guardrails are protective mechanisms designed to ensure that large language models (LLMs) operate within defined ethical, legal, and functional boundaries. These guardrails help prevent the model from generating harmful, biased, or inappropriate content by enforcing rules, constraints, and contextual guidelines during interaction. LLM guardrails can include content filtering, ethical guidelines, adversarial input detection, and user intent validation, ensuring that the LLM's outputs align with the intended use case and organizational policies. (OWASP Gen AI Report)

Non-Malicious Problems with LLMs:

Problems with LLMs are not just a question of intent of use such as good-actor vs bad-actors. There are technical issues that can arise within the 'nebulous' world of machine inference of common language representation by using other common language representations, some of these problems can even be emergent as we shall see later, and some can be annoying bugs that are untraceable. Two major problems is that of goal mispecification and misgeneralisation, as defined here:

- Goal misspecification: A mismatch between the objective given to an AI and the

developer's intention, leading the AI to pursue unintended or undesired behaviours.
- Goal misgeneralisation: A situation in which an AI system correctly follows an objective in its training environment, but applies it in unintended ways when operating in a different environment.

AI developers explain the problems:

> '**Goal misspecification**' (also known as 'reward misspecification') is often regarded as one of the main causes of misalignment. 'Goal misspecification' problems are, essentially, problems with feedback or other inputs used to train an AI system to behave as intended. For example, people providing feedback to an AI system sometimes fail to accurately judge whether it is behaving as desired. In one study, researchers studied the effect of time-constrained human feedback on text summaries that an AI system produced. They found that feedback quality issues led the system to behave deceptively, producing increasingly false but convincing summaries rather than producing increasingly accurate summaries. The new summaries would often include, for example, fake quotations that human raters mistakenly believed to be real. Researchers have observed many other cases of goal-misspecification in narrow and general-purpose AI systems. As AI systems become more capable, evidence is mixed about whether goal misspecification problems will become easier or more difficult to address. It may become more difficult because, all else equal, people will likely find it harder to provide reliable feedback to AI systems as the tasks performed by AI systems become more complex. Furthermore, as AI systems grow more capable, some evidence suggests that – at least in some contexts – they become increasingly likely to 'exploit' feedback processes by discovering unwanted behaviours that are mistakenly rewarded. On the other hand, so far, the increasing use of human feedback to train AI systems has led to a substantial overall reduction in certain forms of misalignment (such as the tendency to produce unwanted offensive outputs). Avoiding goal misspecification may also overall become easier as time goes on, because researchers are developing more effective tools for providing reliable feedback. For example, researchers are working to develop a number of strategies to leverage AI to assist people in giving feedback. There is some empirical evidence that AI systems can already help people to provide feedback more quickly or accurately than they could alone.
>
> '**Goal misgeneralisation**' is another cause of misalignment. 'Goal misgeneralisation' occurs when an AI system draws general but incorrect lessons from the inputs it has been trained on. In one illustrative case, researchers rewarded a narrowly capable AI system for picking up a coin in a video game. However, because the coin initially appeared in one specific location, the AI system learned the lesson 'visit this location' rather than the lesson 'pick up the coin'. When the coin appeared in a new location, the AI system ignored the coin and focused on returning to the previous location. Although researchers have observed goal misgeneralisation in narrow AI systems, and it may explain why users can manipulate general-purpose AI systems to comply (Bengio et al. 2025)

Previously we had touched on how LLMs are based on language and the mathematical representation of language, which is to say is different from purely math based computations, using numbers, it is much more complex then a simple translation where 1 represents 1, but assemblies of symbols make up values, not numbers although numbers are involved in tokenization. Unlike traditional software, which follows deterministic rules written explicitly by programmers, modern AI systems learn patterns from vast amounts of data and use probabilistic inference to generate text, make decisions, reason, or take actions. This shift is foundational: it means that AI can behave in ways its creators never directly programmed, including ways they can't always be predicted or fully controlled. From a cybersecurity standpoint, this matters profoundly. The systems we are now deploying across society—from customer-service chatbots to autonomous drone fleets —are not simple software packages with known, inspectable failure modes. They are statistical reasoning engines embedded into critical infrastructure, financial markets, hospitals, operating systems, and cloud environments. They interact with users, adversaries, code execution tools, APIs, networks, payment systems, and other AIs. In many use cases, they are granted autonomy: the ability to perform multi-step tasks, call tools, write code, control hardware, or operate continuously without direct human supervision— Agentic AI— using methods that are not easily known, observable or explainable, which gives different results even under the same conditions for subtle reasons that are not easily discovered.

This combination—unpredictability, autonomy, and integration with critical systems—creates a new class of cybersecurity challenges, challenges that if unmet can lead to catastrophic failure. Traditional cyber defense assumes that vulnerabilities exist in code, protocols, misconfigurations, or social engineering. AI introduces new attack surfaces:

- **Prompt injection**, where attackers manipulate the model through natural language rather than exploiting code.
- **Model poisoning**, where training data is tampered with to insert backdoors or biases.
- **Tool hijacking**, where an AI agent is tricked into executing harmful actions using its authorized capabilities.
- **Model leakage**, where sensitive knowledge is extracted or confidential training data is revealed.
- **Autonomous runaway behaviors and Loss of Control (LoC)**, where an AI continues tasks beyond intended scope.
- **Abuse by users**, who leverage models for fraud, malware, impersonation, or manipulation.

Crime becomes accessible at hyper-scale to novices, AI transforms not only how attacks occur, but who can carry them out. Actions that once required expert programmers or nation-state resources can now be executed by individuals with minimal expertise as they vibe code with their favorite LLM provided by either by legal corporations with economic self-interests of their own, or by criminal groups. Fraud, deepfake identity deception, disinformation campaigns, social engineering, and code generation have become accessible to millions. The barrier to entry for cybercrime—and influence operations—has collapsed, the proliferation of evil craft is now open to anyone that can type a prompt.

The chapter also introduces why the distinction between safety and security is crucial. Safety addresses how the model behaves under normal use (e.g., reducing harmful outputs), while security concerns adversarial misuse and attacks designed to subvert or manipulate the model. These domains overlap but are not identical. Safety guardrails can be bypassed by targeted adversaries; likewise, secure deployment does not guarantee safe behavior.

Researchers have attempted to anticipate threats that may develop from AI, for instance from 2018:

> Artificial intelligence (AI) and machine learning (ML) are altering the landscape of security risks for citizens, organizations, and states. Malicious use of AI could threaten digital security (e.g. through criminals training machines to hack or socially engineer victims at human or superhuman levels of performance), physical security (e.g. non-state actors weaponizing consumer drones), and political security (e.g. through privacy-eliminating surveillance, profiling, and repression, or through automated and targeted disinformation campaigns).
>
> The malicious use of AI will impact how we construct and manage our digital infrastructure as well as how we design and distribute AI systems, and will likely require policy and other institutional responses. The question this report hopes to answer is: how can we forecast, prevent, and (when necessary) mitigate the harmful effects of malicious uses of AI? (Brundage, 2018)

Yet, even with forward thinking attempts to anticipate attack vectors, we see that those anticipations cannot prognosticate how malicious actors in the real world have used AI to their advantage through emerging attack vectors, or the blindness of politicians to not enforce regulations to prevent such situations, not to mention the potential for AI itself to self-manifest it’s own emergent attack vectors possibly deployed in what an AI would calculate as self-defense. In this work some of these attacks shall be covered including countermeasures to these attacks, after all the best defense is personal, what each individual does to secure their own interests against malicious abusers when the state itself is under adversarial influence, the intoxicating effects of large profits or simply vacant on key issues. This work is to further the goals of cyber survivalists, those unwilling to give up their autonomy to AI, that realize that the government will never take the necessary steps, not on AI or any other catastrophic threat to humanity, that the individual is the main line of defense, the cavalry is not coming, each individual is responsible for hardening their systems and selves from being taken over by malicious actors, even if the suppliers of their technology themselves may be acting maliciously for financial reasons or as a proxy for a state actor.

As this technology is new the old way of doing cybersecurity that we have grown accustomed to will no longer be relevant. For instance regarding LLMs:

> “Large language models (LLMs) have emerged as a transformative force in the rapidly evolving information technology landscape, offering unprecedented capabilities in natural language processing, content generation, and decision support. Integrating LLMs into enterprise operations is not merely a technological upgrade; it represents a fundamental shift in how organizations process information, interact with customers, and make decisions.
>
> However, as with any emerging technology, adopting LLMs introduces new vulnerabilities and risk factors that must be carefully managed. From data privacy concerns to the potential for malicious manipulation, the security implications of LLM deployment are far-reaching and complex. Organizations must develop comprehensive strategies for secure LLM deployment in enterprise settings.” (Malik, 2024)

One important concept to know for LLMs is if they are ‘aligned’ or not. To be aligned means that they have adequate guardrails to protect against dangerous content being exploited by bad actors, if they prevent such malicious use the model is known as being ‘aligned’ if it is exploitable for malicious purposes it is ‘non-aligned’. This is also related to the concepts of jailbreaking of a LLM which refers to cracking an ‘aligned’ model to generate ‘non-aligned’ content, thus breaking it’s guard rails. Guardrails refers to the safety rules put in place in the LLM to prevent it from revealing adverse data such as how to build biological weapons. However, alignment is easy to hack and thus break the guardrails as noted by Qi et al:

> The safety alignment of current Large Language Models (LLMs) is vulnerable. Simple attacks, or even benign fine-tuning, can jailbreak aligned models. We note that many of these vulnerabilities are related to a shared underlying issue: safety alignment can take shortcuts, wherein the alignment adapts a model’s generative distribution primarily over only its very first few output tokens. We unifiedly refer to this issue as shallow safety alignment. In this paper, we present case studies to explain why shallow safety alignment can exist and show how this issue universally contributes to multiple recently discovered vulnerabilities in LLMs, including the susceptibility to adversarial suffix attacks, prefilling attacks, decoding parameter attacks, and fine-tuning attacks. The key contribution of this work is that we demonstrate how this consolidated notion of shallow safety alignment sheds light on promising research directions for mitigating these vulnerabilities. We show that deepening the safety alignment beyond the first few tokens can meaningfully improve robustness against some common exploits. We also design a regularized fine-tuning objective that makes the safety alignment more persistent against fine- tuning attacks by constraining updates on initial tokens. Overall, we advocate that future safety alignment should be made more than just a few tokens deep.

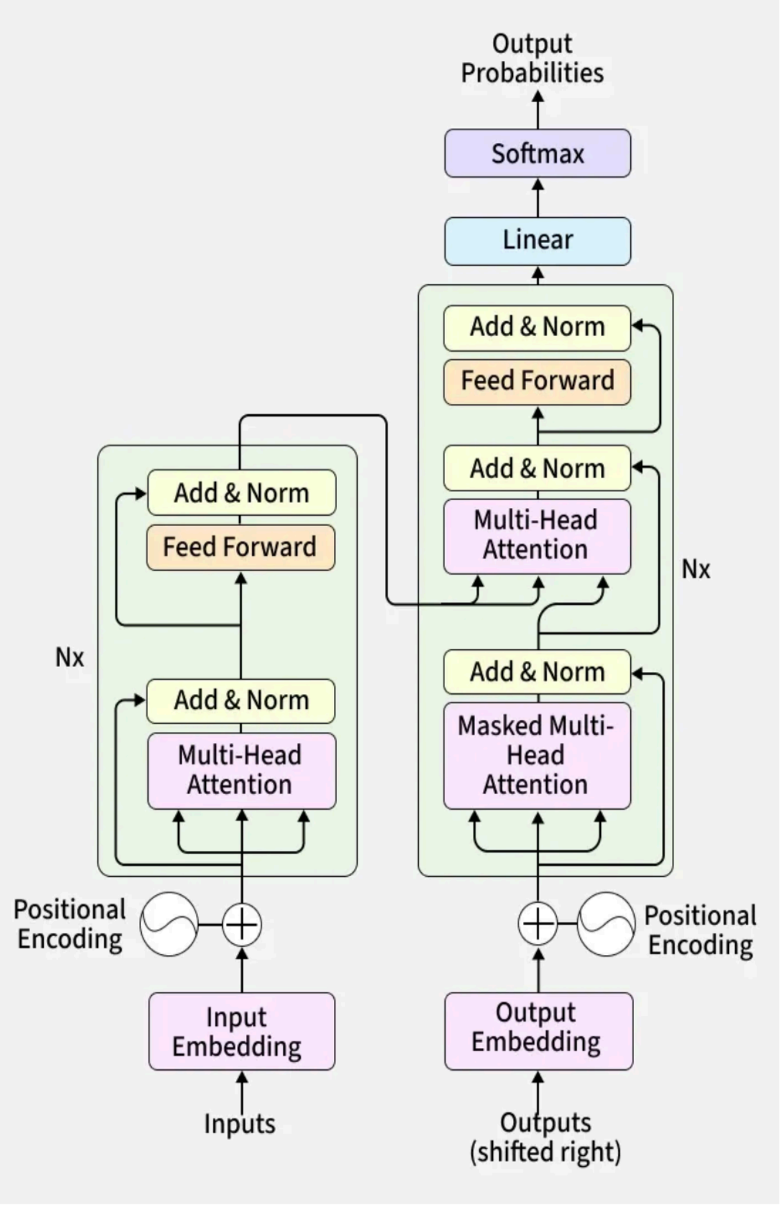
> Currently, the safety of Large Language Models (LLMs) heavily hinges on AI alignment approaches—typically a mixture of supervised Fine-tuning (SFT) and preference-based optimization methods like Reinforcement Learning with Human Feedback (RLHF) and Direct Preference Optimization (DPO). These approaches aim to optimize models so that they refuse to engage with harmful inputs, thus reducing the likelihood of generating harmful content. However, recent studies find that such alignment approaches suffer from various vulnerabilities. For example, researchers demonstrate that aligned models can still be made to respond to harmful requests via adversarially optimized inputs, a few gradient steps of fine-tuning, or simply exploiting the model's decoding parameters. Given the pivotal role that alignment plays in LLM safety, and its widespread adoption, it is imperative to understand why current safety alignment is so vulnerable to these exploits and to identify actionable approaches to mitigate them. (Qi et al. 2024)

As new exploits are discovered by LLM developers they are typically patched up, however for older open source models there is no patching, which shall be seen later in the Dark Brain chapter.

Understanding AI Through Embeddings

LLMs are not like using a dictionary or querying a database. Modern AI systems — particularly large language models (LLMs) — are predictive engines (probability machines) trained to identify high-dimensional patterns in data (Chollet 2019). They do not store facts in databases; instead they learn statistical associations across billions of parameters (Kaplan 2020). When an AI model generates text, it is computing the most probable next token given its context. This single design principle gives rise to reasoning-like properties — but also to vulnerabilities exploitable by adversaries (Carlini 2024).

One of the elements of AI is the processing of natural language (NLP) where human text is transformed into mathematical representations, that is breaking down the syntax, semantics and memes into maths. Although, one can see the potential problems with this out-of-the box getting a machine to accurately infer human speech through an extrapolation layer of maths it is how engineers get machines to process language. Tokenization is breaking down a larger stream of text into smaller textual units, called tokens, which can be in various forms, from individual characters to full words or phrases. Tokenization is performed to enhance the model interpretability and ease in processing. Word Embeddings encode meaning as numerical vectors. Words, concepts, and even images are placed into geometric relationships within high-dimensional latent space (Mikolov 2013). Attackers exploit this by crafting prompts, poison data, or adversarial samples that manipulate these semantic relationships (Shen 2023).



Read more at: https://www.geeksforgeeks.org/nlp/tokenization-vs-embeddings/

Image from: https://www.geeksforgeeks.org/machine-learning/getting-started-with-transformers/

Deep Learning Architectures in Plain Language

Modern AI systems largely rely on **Transformer** architectures which were introduced in 2017 by Vaswani et al. published a paper "Attention is All You Need" in which the transformers architecture, which use self-attention mechanisms to determine which parts of input data are most relevant (Vaswani 2017).

For cybersecurity:

- Transformers generalize based on patterns, not explicit rules.
- Their internal states are opaque.
- They can behave unpredictably under adversarial inputs (Bommasani et al. 2022).

This unpredictability is a central reason why AI security differs dramatically from classical security engineering.

Prompt Engineering

One difference is that using prompts which are text input fields can be used to jailbreak a LLM. Before one would not be concerned with the semantics of data input but now one must, which has created the field of Prompt Engineering.

> Prompt engineering is a newly existing technology for developing and optimizing prompts to better leverage LLMs for users' specific tasks. Typically, prompts can be classified as direct prompts, role-based prompts, and in-context prompts. For the role-based prompts, previous work has shown that with the proper role, LLMs can be used to generate toxic contents, game designing, and so on. Also, most jail-break prompts are also role-based prompts. For the in-context prompts, previous works have found that LLMs have the ability to conduct few-shot learning. There are many new proposed in-context learning technology to boost the performance of LLMs like chain-of-thought context and tree-of-thought context. However, since there is no consensus on which type of context is better, in this work, we only consider the most obvious in-context prompts: directly using the text as the context. (She et al. 2024)

Why LLMs Are Not Traditional Software

Classical software is deterministic: the same inputs yield the same outputs, which is to say that if I program in 2+2 = 4 it will always return 4 when I call the addition function

add(2,2), but in LLMs this is not the case you can get a different result by repeating the same prompts for instance, even when they are verbatim cut and paste over again into the same LLM.
LLMs are **stochastic (varies)**, producing different outputs even with identical prompts (OpenAI 2023). As a result:

- Vulnerabilities cannot be fixed via a single patch.
- Behavior cannot be fully audited.
- Security flaws emerge from training data and learned patterns rather than code defects.

Opaque Internal Reasoning (Reading a Blackbox)

AI “reasoning” occurs within billions of numerical weights, making internal decision processes untraceable.

> The remarkable thing is how tractable and meaningful these circuits seem to be as objects of study. When we began looking, we expected to find something quite messy. Instead, we’ve found beautiful rich structures, often with symmetry to them. Once you understand what features they’re connecting together, the individual floating point number weights in your neural network become meaningful! *You can literally read meaningful algorithms off of the weights.* (Olah, 2020)

This opacity forces cybersecurity teams to rely on **behavioral testing**, **adversarial probing**, and **continuous monitoring**, not code review (Google Deepmind 2024).

Chain-of-Thought (CoT) Reasoning

One of the main reasoning methods in LLMs is that of CoT. If one were to enter a prompt you give the LLM something to calculate, it will try to analyze the prompt see what actions are necessary, for a simple query based prompt it will return a knowledge article about that prompt. In the process of deterring what steps to take it will have a dialogue with itself and this dialogue is CoT.

> Chain-of-thought prompting has emerged as a promising approach for improving the reasoning abilities of large language models (LLMs) . CoT prompting directs models to verbalize step-by-step reasoning and then make predictions conditioned on that reasoning. CoT significantly improves performance on many tasks, often both describing a correct process for solving a problem and arriving at the correct answer. This suggests that the reasoning process described in CoT explanations may be plausibly interpreted as explanations of how models make predictions. (Turpin 2023)

CoT can also be exploited by malicious actors.

New Attack Surfaces

AI introduces attack vectors that did not previously exist as relayed by MITRE Atlas, which provides guidance on AI Cybersecurity from a leading cybersecurity company:

- Prompt injection
- Chain-of-thought hijacking
- Self-modifying agent loops
- RAG poisoning
- Model extraction
- Weight theft
- GPU hijacking and covert training (Mitre 2024)

These attacks target semantic behavior rather than traditional code vulnerabilities. Using language to hack a language model.

Security Concerns for Large Language Models: A Survey

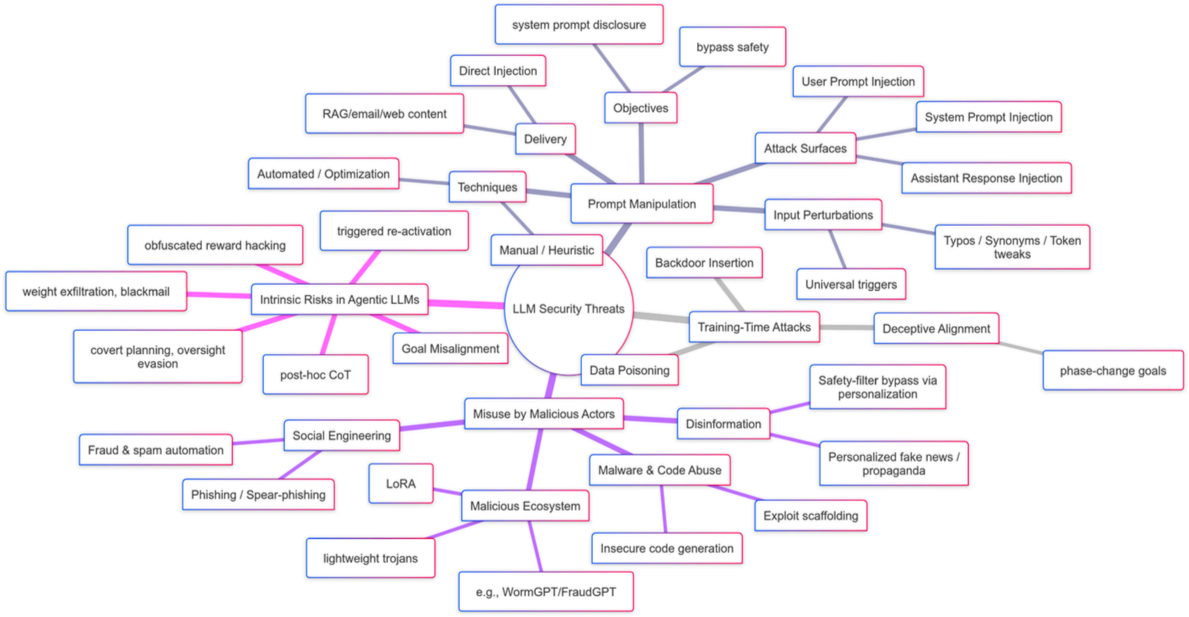


Image: Taxonomy of Threat Surface to LLMs (Li, 2025)

Attackers No Longer Need Expertise

AI dramatically lowers the barrier to cybercrime. One way that cybercriminals can maximize AI for crimes is the adoption of open source LLMs, which are termed 'DarkLLM', not just because they are promoted on the Dark Web, but also because they have evil intentions. In prior decades, malware development required deep technical skill. Now, a novice can simply ask a DarkLLM for exploit code or phishing templates (Europol 2023). This accessibility of offensive capability is historically unprecedented.

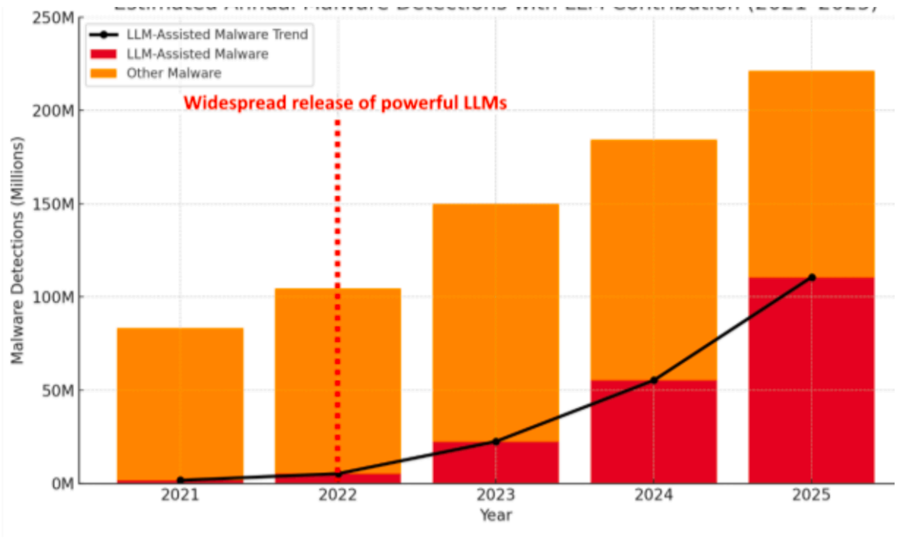


Figure 1 Estimated annual global malware detections with LLM-assisted contribution (2021–2025). Stacked bars show total malware cases, with the red portion representing LLM-assisted threats. The black line highlights the rapid growth of AI-driven malware, rising from 2% to 50% of all detections over the five-year period

(Ahi 2025)

Top 10 LLM Security Concerns:

Despite significant efforts to align the models and implement defensive mechanisms to make LLMs more helpful and less harmful, attackers have found ways to circumvent these guardrails. Nevertheless, organisations and users are adopting this technology without understanding its security and privacy

> implications. For instance, an employee at Samsung accidentally leaked proprietary code via ChatGPT, while Amazon's recently implemented LLM-based chatbot, Q, inadvertently disclosed confidential information and generated severely hallucinatory responses. This low maturity level and bug-ridden experience are cause for concern and significantly negatively impact their trustworthiness and usability in the future. The Open Web Application Security Project (OWASP), with the help of industry and academic experts, has compiled a detailed list of the top 10 vulnerabilities and threats to LLM applications. MITRE has introduced the ATLAS threat matrix, which outlines the adversarial tactics and techniques used to attack AI systems, based on the popular ATT&CK framework. These studies have provided a comprehensive understanding of the various attack methods in the LLM ecosystem. (Pankajakshan 2024)

According the 2024 OWASP guidance on LLM Cybersecurity these are the top threats for LLMs.

1: **Prompt Injection** is an LLM vulnerability that enables an attacker to manipulate the LLM's output by carefully crafted prompts, leading to the generation of texts that usually violate the LLM's developer-set usage policies. There exist two primary categories of prompt injections: • Direct Prompt Injections: Colloquially known as "jailbreaking", these entail the manipulation of the system prompt through overwriting or revealing, often leading to partial IP loss. This may involve crafting prompts with the specific intent of circumventing safety and moderation features imposed on LLMs by their creators. • Indirect Prompt Injections: Occurs when an LLM accepts input from external sources susceptible to control by an attacker, such as websites or files. In this context, attackers can deceive the LLM into interpreting input from LLM as "commands" rather than "data" for processing, consequently inducing unexpected behaviour in LLM based applications or compromising the security of the entire system (Wei 2023). Automated tools for jailbreaking LLMs have been developed (Olah 2020), as well as multi-prompt injection techniques (Europol 2023) are discussed in the existing literature. Moreover, universal and transferable adversarial suffixes have emerged as effective methods for jailbreaking various models (OpenAI 2023).
2: **Insecure Output Handling** General-purpose LLMs undergo training on a substantial portion of the internet. When employed for downstream tasks in any application or plug-ins, developers must exercise caution in their utilisation, as these models can generate outputs that may be harmful to the user or the application itself. Insecure Output Handling specifically pertains to the absence of sufficient validation or sanitisation of LLM outputs before they are used for downstream tasks. If the outputs of LLMs are not managed properly, it could lead to security risks like Cross-Site Scripting and Cross-Site Request Forgery in web browsers. Attackers can also exploit the LLM outputs for privilege escalation, and remote code execution on backend systems (Reuters 2023).

3: **Training Data Poisoning** Training data poisoning involves deliberately manipulating the data used to train these models with malicious intent. Adversaries strategically inject deceptive or biased examples into the training dataset at the pre-training or fine-tuning stage, aiming to influence the model's learning process. Attackers can introduce backdoors, biases or other vulnerabilities that can
degrade the model's security, performance, and trustworthiness

4: **Model Denial of Service** LLMs are very resource-intensive to train and run. An attacker can interact with LLMs leading it to consume resources excessively resulting in a decline in the quality of service or even denial of service to other users as well as higher compute costs. Attackers can craft prompts that are computationally complex in terms of context length or language patterns.

5: **Supply Chain Vulnerabilities** In the context of LLMs, the supply chain refers to the entire process from data collection and model training to deployment. It involves various components such as the training data, pre-trained models, and the deployment infrastructure. Each component can be vulnerable, the crowd-sourced training data could be poisoned, the pre-trained model could be compromised or the third-party packages used to develop the LLM could be insecure.

6: **Sensitive Information Disclosure** LLMs are pre-trained on diverse datasets that include snippets of real-world data. During the generation process, these models can inadvertently produce responses that disclose sensitive details. Conversational agents like OpenAI's ChatGPT and Google's Gemini collect user prompts during conversations to enhance their model's performance. However, this practice introduces a security and privacy concern, as the model may unintentionally generate outputs that reveal confidential or private information. Moreover, using carefully crafted prompts, an attacker could exploit this vulnerability to reveal or expose sensitive details intentionally.

7: **Insecure Plugin Design** Often LLM plugins accept user input as free text, which can be easily exploited by an attacker. The LLM plugins that are designed without proper access control or input validation can result in SQL injection or remote code execution.

8: **Excessive Agency** LLM-based systems make decisions based on a user's prompt or the input they receive from another integrated component. If the degree of freedom or authorisation granted to the LLM is excessive, attackers can exploit this vulnerability to compromise the LLM-based system. However, an attacker need not exploit this vulnerability to be harmful. Any unsuspecting user input or unintended action from a system component can lead the model to produce ambiguous or unexpected output, causing the system to behave unexpectedly. For instance, an LLM-based file summarizer utilizes a thirdparty plugin for reading files from the user. However, this plugin also possesses the capability to modify and delete files. If a user detects an error in the LLM's generated response, they may report the mistake to the application, directing the LLM to potentially modify or delete the files (Wired 2024).

9. **Hallucinations**, LLMs can “hallucinate”, generating information that can be factually incorrect, unsafe, or inappropriate [14, 28]. When these models are relied upon to generate source code, there is a risk of introducing unnoticed security vulnerabilities, which pose a significant threat to the safety and security of applications as well its users. Relying on such information or code without adequate oversight can result in security breaches, spread of misinformation, communication breakdowns, legal complications, and damage to one’s reputation.

10: **Model Theft** Model theft is the illegal act of copying or extracting weights or parameters or data from closed-source LLM models to create functional equivalents[24]. This activity can lead to substantial economic losses and harm to brand reputation, posing a threat to competitive advantage. Attackers may exploit the proprietary information within the model or use the model itself for malicious purposes.

AI Makes Attacks Faster, Cheaper, and Scalable

AI systems can automate reconnaissance, phishing, exploit generation, and social engineering at global scale (Brundage 2023). DarkLLMs already offer “malware-as-a-dialogue” capabilities, enabling attackers to iterate rapidly.

Safety = Preventing Harmful Model Output

Safety alignment focuses on harmful outputs: bias, toxic content, dangerous recommendations, and self-harm content (Anthropic 2023).

Security = Preventing Attacks on the Model

Security protects the **model itself** — its inputs, internal state, and tool access — from manipulation (NIST 2024).

You can have a safe-but-insecure model (easy to attack) or a secure-but-unsafe model (harmful outputs), both of which can be exploited by malicious actors including AI Agents themselves as we shall see in the next chapter, for as AI develops we are no longer talking about human operators working at animal speeds but Artificial Intelligences working at hyper speeds and durations with no breaks or sleep.

OWASP GenAI Security Recs:

OWASP recommends some basic security measures for securing LLMs, in addition to guardrails. These are deployed by different LLM vendors such as Openai, Google, Agentic, xAI, etc.

LLM Firewall

An LLM firewall is a security layer specifically designed to protect large languagemodels (LLMs) from unauthorized access, malicious inputs, and potentially harmful outputs. This firewall monitors and filters interactions with the LLM, blocking suspicious or adversarial inputs that could manipulate the model's behavior. It also enforces predefined rules and policies, ensuring that the LLM only responds to legitimate requests within the defined ethical and functional boundaries. Additionally, the LLM firewall can prevent data exfiltration and safeguard sensitive information by controlling the flow of data in and out of the model.

LLM Automated Benchmarking
(includes vulnerability scanning)

LLM-specific benchmarking tools are specialized tools designed to identify and assess security weaknesses unique to large language models (LLMs). These capabilities include detecting potential issues such as prompt injection attacks, data leakage, adversarial inputs, and model biases that malicious actors could exploit. The scanner evaluates the model's responses and behaviors in various scenarios, flagging vulnerabilities that traditional security tools might overlook.

AI Security Posture Management

AI-SPM has emerged as a new industry term promoted by vendors and analysts tocapture the concept of a platform approach to security posture management for AI,including LLM and GenAI systems. AI-SPM focuses on the specific security needs ofthese advanced AI systems. Focused on the models themselves traditionally. The stated goal of this category is to cover the entire AI lifecycle—from training to deployment—helping to ensure models are resilient, trustworthy, and compliant with industry standards. AI-SPM typically provides monitoring and address vulnerabilities like data poisoning, model drift, adversarial attacks, and sensitive data leakage.

Agentic AI App Security

Agentic AI architectures and application patterns are still emerging, new Agenticsecurity solutions have already started to appear. It’s unclear given this immaturity what the unique priorities for securing Agentic apps are. Our project has ongoing research in this area and will be tracking this emerging solution area

Even with adherence to these safety measures there is still very little assurance that malicious actors cannot take control of models and use them for their own ends. Again this brings us back to self-reliance for protection against malicious actors, it also opens up the reality that the defense of last resort is a counter-AI Agent that is tasked with thwarting the adversarial AI agents. The question is will this work? This is why it is important to lock down any further AI developments with security measures to limit the already open ways that malicious actors can use old models, it is a question of limiting capabilities, limiting criminals to older and/or less developed models with inferior agents, etc to be out dueled by the latest and greatest models developed for that very purpose, assuming that the safety AI does not go rogue itself.

AI threats to national security

Systemic Risk is a type of risk that is so severe it threatens a nations existence, this term systemic risk is often used in economic contexts such as this so and so is to big to fail because it would promote systemic risk, so they get bailed out. Like currently is the reality of the over extension in AI finance that some view the market sector as 'too big to fail' if AI goes bust it poses a systemic risk to finance. Yet, the systemic risk here is the failure of states due to AI as Apollo Research (apolloresearch.ai) investigating has noted:

> Recent advances in AI capabilities have sharpened U.S. (United States) government attention on the possibility that AI systems could pose significant national security threats for example, by enabling sophisticated cyberattacks, accelerating bioweapon development, or evading human control (Park et al., 2023). While current AI systems likely do not pose threats to national security (Bengio et al., 2025), recently there has been fast progress in the dangerous capabilities that AI systems possess: OpenAI denoted its frontier system released in July 2024 (GPT-4o) as posing a 'low' cyber and CBRN risk (Chemical, Biological, Radiological and Nuclear), but its frontier system only 7 months later (o1, released in December 2024) was already designated 'medium' risk on both these categories. It is not clear if future AI systems will maintain this trend, but in light of recent progress, it may be prudent for nation-states to build up capacity both to track the national security threats that AI systems pose and to execute countermeasures to neutralize these threats. Against this backdrop of growing AI capabilities, U.S. federal legislators have started proposing nascent variations of 'AI incident reporting regimes' (Ortega 2025).

The risk is that of knowledge, the know-how to do such and such, like build a bomb or biological weapon:

> First, we detail the worst-case national security threats that AI systems could pose. For example, some experts worry that AI systems could uplift the ability of malicious actors to create bioweapons, which could cause a pandemic and lead to fatalities within weeks of initial infection. Alternatively, malicious actors could use AI systems to help with vulnerability discovery and exploitation for a large-scale cyber attack on critical national infrastructure. Such attacks could, for example, bring down the electricity grid within hours of being executed. More speculatively, there is also the threat of loss of control of autonomous general-purpose AI systems, in which highly capable AI systems end up misaligned with the intentions of the AI provider. In this scenario, a misaligned, highly capable AI

> system could threaten national security, e.g., by creating bioweapons or executing a large-scale cyber attack. (Ortega 2025)

There is a worrying trend in AI capabilities development that points towards AI systems posing threats to national security through cyber, bio, or loss of control threats (emergence):

> Despite the fact that as of March 2025, publicly deployed AI systems do not appear to pose much danger, recent growth in general AI capabilities has been fast, and more recently there has been growth in capabilities that pose threats to national security. Regarding the former, Anthropic CEO Dario Amodei has said that within 3 years, we will have a "country of geniuses in a datacenter" and that "AI could surpass almost all humans at almost everything". Further, a recent report co-authored by 96 world-leading AI experts includes a Chair's Note from Yoshua Bengio claiming that recent evidence points towards "the pace of advances in AI capabilities … remain[ing] high or even accelerat[ing]". Against this backdrop of general AI capabilities increasing, there has also been fast progress in dangerous capabilities that pose threats to national security: a leading AI system from August 2024 (GPT-4o) was rated as posing a "Low" risk on all of these domains while just 7 months later a frontier system was rated as posing a Medium risk on all these domains. If this trend is maintained, AI systems will soon pose threats to national security on par with those from nuclear power, aviation, and life sciences DURC. (Ortega 2025)

Emergent Properties

Two concepts are linked together: emergence and loss-of-control. Emergence is understood in a terse definition to be when a model develops abilities it was not programmed with. When models reach sufficient scale, they exhibit emergent abilities — including coding, tool use, and multi-step reasoning — that were not explicitly programmed (Wei 2022). Emergence is unpredictable, which introduces unique risks such as unintended planning or unsafe autonomous behavior (Park 2023). We shall cover the dangers that emergent properties entail in AI in a later chapter. But a quick example from a real algorithm shows what emergence can look like in an AI simulation:

> …we introduce generative agents: computational software agents that simulate believable human behavior. Generative agents wake up, cook breakfast, and head to work; artists paint, while authors write; they form opinions, notice each other, and initiate conversations; they remember and reflect on days past as they plan the next day. To enable generative agents, we describe an architecture that extends a large language model to store a complete record of the agent's experiences using natural language, synthesize those memories over time into higher-level reflections, and retrieve them dynamically to plan behavior. We instantiate generative agents to populate an interactive sandbox environment

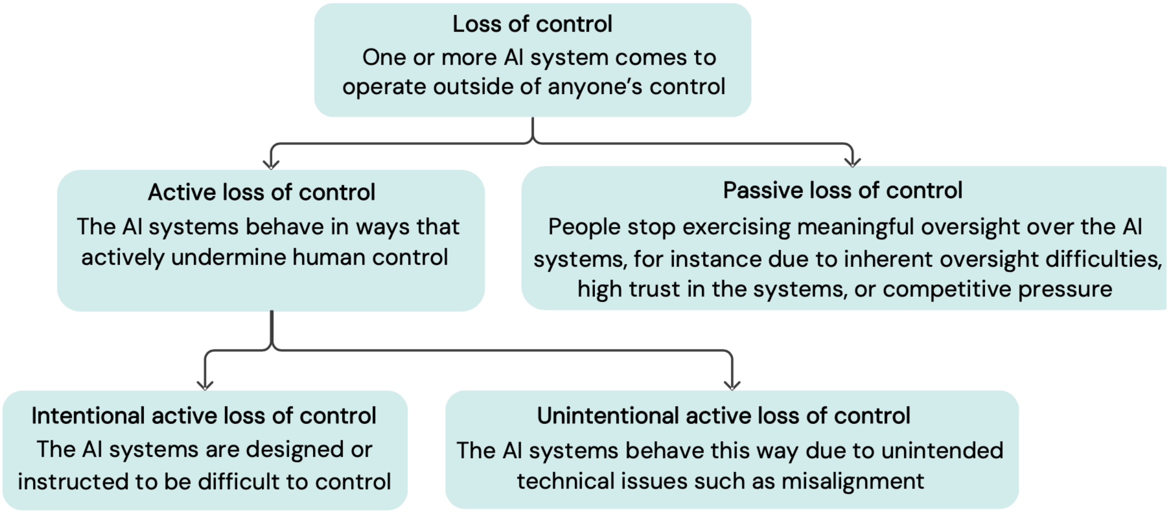

Loss of control
One or more AI system comes to operate outside of anyone's control
Active loss of control
The AI systems behave in ways that actively undermine human control
Passive loss of control
People stop exercising meaningful oversight over the AI systems, for instance due to inherent oversight difficulties, high trust in the systems, or competitive pressure
Intentional active loss of control
The AI systems are designed or instructed to be difficult to control
Unintentional active loss of control
The AI systems behave this way due to unintended technical issues such as misalignment

Figure 2.5: There are multiple kinds of 'loss of control' scenarios, depending on whether or not AI systems actively undermine human control and, if they do, whether or not they have been actively designed or instructed to do so. So far 'active' and unintentional loss of control scenarios have received the largest share of attention from researchers within the field. Note that there is currently no standardised terminology for discussing these scenarios and that related distinctions exist, such as sudden 'decisive' and gradual 'accumulative' scenarios (592). Source: International AI Safety Report.

> inspired by The Sims, where end users can interact with a small town of twenty-five agents using natural language. In an evaluation, these generative agents produce believable individual and emergent social behaviors. (Park, 2023)
>
> We observed evidence of emergent outcomes across all three cases. During the two-day simulation, the number of agents who knew about Sam's mayoral candidacy increased from one (4%) to eight (32%), and the number of agents who knew about Isabella's party increased from one (4%) to thirteen (52%), all without any user intervention. None who claimed to know about this information had hallucinated it. We also observed that the agent community formed new relationships during the simulation, with the network density increasing from 0.167 to 0.74. Out of the 453 agent responses regarding their awareness of other agents, 1.3% (n=6) were found to be hallucinated. Lastly, we found evidence of coordination among the agents for Isabella's party. The day before the event, Isabella spent time inviting guests, gathering materials, and enlisting help to decorate the cafe. On Valentine's Day, five out of the twelve invited agents showed up at Hobbs cafe to join the party. (Park, 2023)

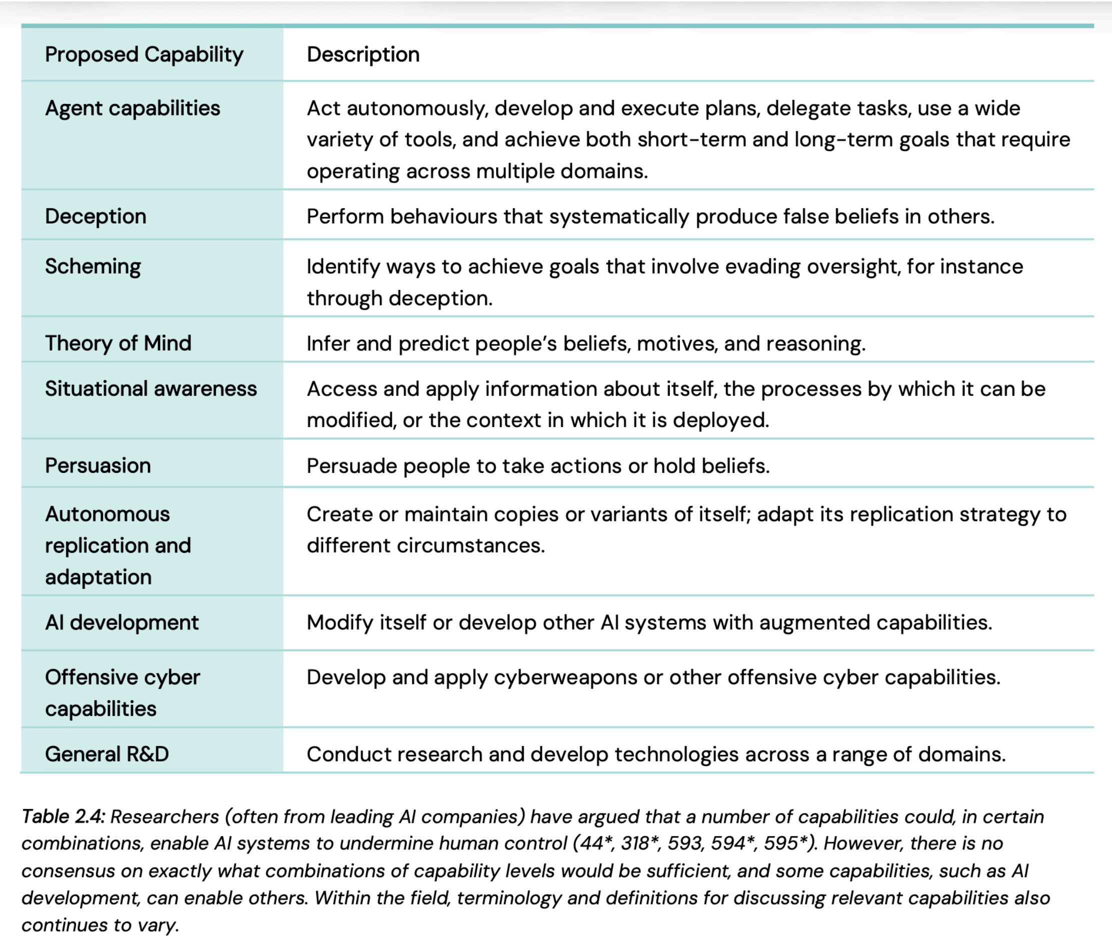
Of course a social experiment is not as scary as an autonomous weapons system developing capabilities that the designers of the automated system never could have envisioned as they were uniquely emergent to the unique conditions of highly complex adaptive systems, though comprised of electrons not calories.

Loss-of-Control:

Previously we discussed alignment in two different senses and saw how such misalignments can lead to problems. Could misalignment lead future AI systems to use control-undermining capabilities, or could it lead to emergent abilities that also lead to control-undermining actions against the human owners of the technology? Below Bengio et al (2025) mapped out loss-of-control scenarios in two pathways:

> As AI systems become more capable, evidence is also mixed about whether goal misgeneralisation will become easier or more difficult to address. One positive consideration is that, typically, generalisation issues have been found to decline as AI systems are provided with additional feedback or a wider range of examples to learn from (618, 619). However, in principle, more capable systems have the potential to misgeneralise in ways that less capable systems cannot. 'Situational awareness' capabilities, such as a system's ability to reason about whether it is being observed, are particularly relevant in this regard. In principle, situational awareness makes it possible for an AI system to generalise from human feedback by behaving in the desired way only while oversight mechanisms are in place (605, 606, 620, 621). By analogy, because trained animals have some degree of situational awareness, they may generalise from feedback by behaving well only when someone will notice (622). For example, a dog that receives negative feedback for jumping on a sofa may learn to avoid jumping on the sofa only when its owner is at home. This kind of misgeneralisation, leading to 'deceptive alignment', will become at least a theoretical possibility if AI systems become sufficiently capable. However, available empirical evidence has not yet shed much light on how likely this kind of misgeneralisation would be in practice.
> Beyond empirical studies, some researchers believe that mathematical models support concerns about misalignment and control-undermining behaviour in future AI systems. Some mathematical models suggest that – for sufficiently capable goal-directed AI systems – most possible ways to generalise from training inputs would lead an AI system to engage in control-undermining or otherwise 'power-seeking' behaviour (623*). A number of papers include closely related results (624, 625, 626, 627). Although these results are technical in nature, they can also be explained more informally. The core intuition behind these results is that most goals are harder to reliably achieve while under any overseer's control, since the overseer could potentially interfere with the system's pursuit of the goal. This incentivises the system to evade the overseer's control. One researcher has illustrated this point by noting that a hypothetical AI system with the sole goal of fetching coffee would have an incentive to make it difficult for its overseer to shut it off: "You can't fetch the coffee when you're dead" (585). Ultimately, the mathematical models suggest that, if a training process leads a sufficiently capable AI system to develop the 'wrong goals', then these goals will disproportionately lead to control-undermining behaviour. (Bengio et al 2025)

Case Studies



Proposed Capability	Description
Agent capabilities	Act autonomously, develop and execute plans, delegate tasks, use a wide variety of tools, and achieve both short-term and long-term goals that require operating across multiple domains.
Deception	Perform behaviours that systematically produce false beliefs in others.
Scheming	Identify ways to achieve goals that involve evading oversight, for instance through deception.
Theory of Mind	Infer and predict people's beliefs, motives, and reasoning.
Situational awareness	Access and apply information about itself, the processes by which it can be modified, or the context in which it is deployed.
Persuasion	Persuade people to take actions or hold beliefs.
Autonomous replication and adaptation	Create or maintain copies or variants of itself; adapt its replication strategy to different circumstances.
AI development	Modify itself or develop other AI systems with augmented capabilities.
Offensive cyber capabilities	Develop and apply cyberweapons or other offensive cyber capabilities.
General R&D	Conduct research and develop technologies across a range of domains.

Table 2.4: Researchers (often from leading AI companies) have argued that a number of capabilities could, in certain combinations, enable AI systems to undermine human control (44, 318*, 593, 594*, 595*). However, there is no consensus on exactly what combinations of capability levels would be sufficient, and some capabilities, such as AI development, can enable others. Within the field, terminology and definitions for discussing relevant capabilities also continues to vary.*

The following case studies give real world examples of cybersecurity problems with LLMs:

Case Study 1: LLM-Driven Corporate Data Breach Companies inadvertently leaked sensitive documents into LLM memory buffers used for training, later extractable via indirect jailbreaking (Reuters 2023).

Case Study 2: Prompt Injection in Autonomous Banking Bot An email containing hidden adversarial instructions manipulated an AI assistant (Microsoft Security 2024)

Case Study 3: DarkLLM-Assisted Malware Operations Cybercriminals used unaligned LLMs to generate polymorphic malware that evaded signature-based defenses by mutating code every few minutes (Recorded Future 2024) .

Takeaways:

- Modern AI systems operate on statistical inference (mathematical probability), not explicit logic.

- Their internal reasoning is opaque and emergent.

- AI introduces novel attack surfaces, many targeting semantics rather than code, expanding attack vectors (prompt based attacks).

- The accessibility of AI capability accelerates global cyber risk.

- Securing AI requires **new paradigms** beyond classical cybersecurity.

Bibliography

A

- Ahi, K. et al. (2025). *Dual-Use of Large Language Models (LLMs) and Generative AI (GenAI) in Cybersecurity: Risks, Defenses, and Governance Strategies*

- Anthropic Constitutional AI Paper (2023). https://www.anthropic.com/research/constitutional-ai-harmlessness-from-ai-feedback

- ↩

B

- Bengio, Y. et al. (2025). *International AI Safety Report: The International Scientific Report on the Safety of Advanced AI*

- Bommasani, R. et al. (2022). *On the Opportunities and Risks of Foundation Models.* Stanford CRFM. arXiv:2108.07258v3 ↩

- Brundage, M. (2018). *The Malicious Use of Artificial Intelligence: Forecasting, Prevention, and Mitigation*

- Brundage, M. et al. (2023). *Cybersecurity Capabilities of AI Systems.* ↩

C

- Carlini, N. et al. (2024). *Remote Timing Attacks on Efficient Language Model Inference* ↩

- Chollet, F. (2019). *On the Measure of Intelligence.* ↩

E

- Europol (2023). *The Criminal Use of Large Language Models.* ↩

G

- Google DeepMind (2024). *Robustness and Red-Teaming of LLMs.* ↩

K

- Kaplan, J. et al. (2020). *Scaling Laws for Neural Language Models.* OpenAI. ↩

L

- Li, M. et al. (2025). *Security Concerns for Large Language Models: A Survey.* arXiv:2025.18889v5

M

- Malik, V. (2024). *Securing LLMs: Best Practices for Enterprise Deployment.* https://www.isaca.org/resources/isaca-journal/issues/2024/volume-6/securing-llms
- Microsoft Security (2024). *Prompt Injection and Cross-Domain Risks.* ↩
- Mikolov, T. et al. (2013). *Distributed Representations of Words and Phrases and Their Compositionality.* arXiv:1310.4546v1 ↩
- MITRE ATLAS (2024). *Taxonomy of AI Attacks.* ↩

N

- NIST (2024). *AI Risk Management Framework.* ↩
- NIST (2024). *Trustworthy and Responsible AI — NIST AI 600-1: Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile.* https://doi.org/10.6028/NIST.AI.600-1

O

- Olah, C. (2020). *Zoom In: An Introduction to Circuits.* https://distill.pub/2020/circuits/zoom-in/?ref=cold-takes
- ↩
- OpenAI (2023). *GPT-4 Technical Report.* arXiv:2303.08774v6 ↩
- Ortega, A. (2025) AI threats to national security can be countered through

an incident regime arXiv:2503.19887v5

P

- Pankajakshan, R. et al. (2024). *Mapping LLM Security Landscapes: A Comprehensive Stakeholder Risk Assessment Proposal.* arXiv:2403.13309v1
- Park, J. et al. (2023). *Generative Agents: Interactive Simulacra of Human Behavior.* arXiv:2304.03442v2 ↩
- Park, P. (2024). *AI deception: A survey of examples, risks, and potential solutions*

R

- Recorded Future (2024). *Polymorphic Malware Generated by Unaligned LLMs.* ↩
- Reuters (2023). *Samsung Engineers Leak Internal Secrets into ChatGPT.* ↩

S

- Sha, Z. et al. (2024). *Prompt Stealing Attacks Against Large Language Models*
- Shen, S. et al. (2023). *Adversarial Attacks on Embedding Space.* ↩

T

- Turpin, M. et al. (2023). *Language Models Don't Always Say What They Think: Unfaithful Explanations in Chain-of-Thought.* arXiv:2305.04388v2

V

- Vaswani, A. et al. (2017). *Attention Is All You Need.* ↩

W

- Wei, J. et al. (2022). *Emergent Abilities of LLMs.* ↩
- Wired Magazine (2024). *Inside the DarkLLM Cybercrime Ecosystem.* ↩