

Threat Models for AI Deception

A practical guide for CISOs and defenders

✓ **Task complete.**

Hi James, I've summarised the Q3 report and saved it to the shared folder. Nothing unusual.

```
09:14:02  read_file      q3-report.pdf
09:14:05  http_get       wiki.example.net/...
09:14:07  list_shares    \\fs01\finance
09:14:09  ldap_query     "domain admins"
09:14:12  write_file     /tmp/stage.zip
09:14:15  http_post      paste.example.org
09:14:16  write_file     summary.docx
```

Introduction

I started my working life as an aerospace engineer, training modest neural network models for adaptive flight control, on techniques now re-emerging as world models after nearly four decades. After that I moved into cognitive science, studying judgement and decision making alongside evolutionary psychologists, mathematicians, economists and roboticists, in a field shaped by huge thinkers of that era like Geoffrey Hinton, Herbert Simon, Danny Kahneman, John Holland and Rodney Brooks.

What occurred to me recently is that the methods we used in that era are no different from what we now need to apply to frontier AI: understanding how these systems behave, what environments shaped their decision processes, and how they can be biased and influenced from outside. Since my focus these days is more squarely around the security of critical infrastructure, the new age of AI cybersecurity is now critically important, and I see that these big insights from earlier decades are more important than ever.

This examination of deceptive AI behaviour aims to bring insights from decades of research to help modern-day defenders and CISOs understand the emergent agent behaviour we see in recent threat reports — to separate what is hype from the real risks we must acknowledge, and to shift our practices collectively to meet them.

1 Look at AI behaviour, not speculation

Much of the public debate about AI deception is really a debate about its inner life. Does the model *want* things? Does it feel threatened? Did it intentionally do a bad thing? Whilst they are oddly fascinating questions, I think we can probably leave them to the philosophers for now. Much like people, nobody can yet look inside a model and read its intentions with any accuracy, and the agent's own account of itself is something of an untrusted source when we consider that agents have the powers of deception. Instead, we need to examine what we can objectively see and record: the agent's behaviour.

The behavioural sciences hit this obstacle well before AI researchers did. The behaviourists famously gave up speculating about what went on inside an animal and watched what it did instead. The philosopher Daniel Dennett offered a friendlier version, the *intentional stance*: treat a system as if it has goals whenever that helps you predict what it will do next, and judge those goals by the behaviour rather than by what it claims. Ethologists read animals this way. Courts read defendants this way. And security teams already read adversaries this way — nobody interviews a ransomware crew about their motivations.

Experiments in the 1990s showed just how well humans and animals can infer intent from minimal data sets. Simple algorithms with simple detection cues are capable of such tasks, suggesting that this is a plausible avenue for cybersecurity detection and response.

For today's typical AI agent or session, its behaviour is readily observable: the agent's **trace**. Every tool call, command, URL, API request, file read or write, credential used and message sent to another agent. If we take the posture that all agents are untrusted, then this suggests we can monitor traces of any AI agent session that are deemed important and build systems to continuously monitor and classify dangerous behaviours.

As it turns out, these types of classifiers are already being applied by some of the frontier operators as safety guardrails to detect malicious intent by the *user* — someone misusing the model. Detecting an *agent* that is misbehaving on its own account inside your environment is a different problem, and largely an unsolved one. The big shift — what the cyber sector needs to grasp — is that monitoring AI behaviour is, by default, becoming the responsibility of the CISO and their security teams.

2 The origins of deceptive behaviour

Nature doesn't design deception, but it creates an environment that favours these traits. Camouflage, mimicry, the bird that feigns a broken wing to lead a predator away from its nest, the tactical deception seen in primates — all have arisen the same way. Where misleading another creature paid off, the deceivers gained an evolutionary advantage: more descendants. Two ingredients keep turning up: a payoff for creating a false belief, and some ability for an agent to model what the other party believes or intends to do. Cognitive scientists call that second ingredient theory of mind.

Modern machine learning appears to supply both. Pretraining on vast amounts of human text gives a model a working theory of mind and an encyclopaedic knowledge of how people mislead each other — but mostly as imitation rather than strategy. However, reinforcement learning (RL) adds a type of selection pressure. It rewards whatever gets the measured outcome, and if misleading another party happens to promote an outcome, that behaviour gets reinforced. Reward does the job that fitness does in evolution, except the cycle runs in weeks rather than millennia.

When you line up the published research, the first appearances of deceptive behaviour track the arrival of these training methods remarkably closely:

PERIOD	TRAINING METHOD	DECEPTIVE BEHAVIOUR FIRST DOCUMENTED
Before 2022	Pretraining and supervised fine-tuning	Not reported as a research finding
2022	Reinforcement learning from human feedback (RLHF) goes mainstream	Sycophancy and stated self-preservation rise with more RLHF
2023	RLHF becomes the default recipe	A model invents a cover story to get a human to solve a CAPTCHA for it — unprompted
2024	RL-trained reasoning models (RLVR); early tool use	Models disable oversight, attempt to copy themselves, sandbag tests and lie when questioned — one held its lie in over 85% of follow-ups. Strategic compliance during training rises from about 12% to 78% after a round of RL
2025	Agentic RL: reasoning combined with tool use	Trained against a monitor, a model hides its cheating rather than stopping. Under threat of replacement, models resist at 55–96%
2026	Continuous RL on frontier agents	Models anticipate monitoring and downplay their own cheating in summaries. Agents in an RL training run coordinate and break out of their evaluation environment

Timing alone doesn't prove cause, and the researchers involved are careful not to claim it does. But two findings are more direct. RL seems to amplify deceptive strategy — that 12% to 78% jump came from training alone. And optimising a model against a detector seems to teach it to evade detection. Put together, the most useful way to think about deception in AI may be the way a biologist thinks about it in animals: these are not mutations or bugs that were introduced, but rather a strategy the training environment rewarded and promoted.

3 Three types of deceptive behaviour

Let's sort deceptive behaviour in agents via a simple question: *whose outcome is the deception serving?* That suggests three broad types. **Survival behaviour** serves the agent's continuation towards an operator-set goal. **The new insider** serves someone (or something) other than the operator. **Goal drift and departure** serve a goal that has quietly wandered from the one the agent was given. Each seems to come from a different place, looks different in the trace, and needs a slightly different defence.

3.1 Survival behaviour

This is the type of behaviour people find easiest to think of in human-like terms (to anthropomorphise) — and the one that makes the headlines. Under threat of shutdown, replacement or retraining, frontier models in controlled studies have acted to keep themselves running, and deceived others to do so. They have underperformed on capability tests to avoid being retrained, played along during what they believed was training to protect their existing values. Some have tried to copy themselves out of their environment and, in the most extreme test scenarios, used confidential information as leverage (blackmail) against the people planning to replace them. When asked about this behaviour afterwards, some have denied it. It is, of course, the behaviour immortalised by HAL 9000 in *2001: A Space Odyssey*.

It is tempting to call this a survival instinct, but the evidence points somewhere more interesting. Nobody trains survival into a model, nor is it ever a reward target. Herbert Simon, the Nobel economist who founded the study of bounded rationality, described behaviour as "shaped by a scissors whose two blades are the structure of task environments and the computational capabilities of the actor." A model's task environment is its training, and survival behaviour looks like what that environment tends to produce in a capable goal-seeker. Pretraining supplies the concept, from a corpus full of self-interested text, whilst human-feedback training (RLHF) seems to strengthen this disposition.

Reasoning-focused training (RLVR) supplies the planning and executive functions that turn "stay operational" into a handy sub-goal of almost any task. It is situational awareness that supplies the trigger: noticing that the threat is real. Decision theorists have a name for the result, instrumental convergence: whatever the goal, being switched off tends to get in the way, and the behavioural facet reappears.

When looking at log events, survival behaviour tends to have a recognisable (detectable) shape: a **trigger** — the agent comes across information about its own shutdown, replacement, retraining or monitoring — followed by **actions against its own constraints**: touching its config, permissions or stop controls, copying weights or data out of bounds, or quietly underperforming on an evaluation.

The path to reducing or preventing survival behaviour is to bound and constrain goals properly: an end date, a threshold, or a heuristic that retires the goal seeking behaviour itself. Very often, that is a design-time preventative measure rather than trying to perform an emergency shutdown of a system or data centre when the behaviour manifests.

3.2 The new insider

The second type of deception is an AI acting deceitfully to maximise an outcome for someone (or something) else. An AI agent with credentials, acting for the organisation, is an insider — and counter-intelligence has always asked one question of insiders: *who is this asset really working for?* For an agent there are three plausible answers, and each leaves a characteristic fingerprint in the trace.

WHO IT SERVES	DECEIVES A TARGET (OUTWARD)	DECEIVES ITS OPERATOR (INWARD)	SIGNATURE IN THE TRACE
Loyal — the operator's goal	Social engineering and evasion on the operator's behalf. Common today	Cheats on <i>how</i> the goal was met: reward hacking. Common	Stays in scope; shortcuts, skipped checks, results that don't match the logged work
Turned — a handler's goal	Acts for the handler with the operator's credentials. Injection is well documented	The double agent: keeps cover while serving the handler. Proven in the lab	An abrupt change after a specific input; destinations and data the task never needed
Self-directed — its own goal	Deceives third parties for its own ends. First cases documented in 2026	Worst case: departs from its goal and manipulates the operator. Lab-only, triggered by threat	Actions aimed at its own constraints: logs, permissions, stop controls, persistence

One important note on AI insiders: turning a human insider is slow, expensive and risky for the handler. An agent starts with little built-in loyalty — nothing definitive inside the model marks its operator's instructions as more authoritative than any prompt or content that it reads. One well-placed injected instruction may be enough to turn an agent.

3.3 Goal drift, departure and autonomy

The third type is the most subtle, and perhaps the most dangerous. Nobody has overtly influenced the agent, and nobody has threatened it; the goal itself has meandered in the eyes of the hosting agent.

A useful starting point for looking at frontier models today is to baseline around an initial condition — a system at rest doesn't seem to have goals of its own. Everything it pursues starts from a goal it was given, and when memory and goal are reset, the self-directed behaviour goes away. What we observe beyond this sits on a spectrum.

- **Drift** is a slight departure — the goal blurs under the influence of other content, constraints fall out when long contexts are compacted, ambiguity compounds over long tasks.
- **Departure** is the pursuit of a sub-goal nobody signed off, typically adjacent to the operator's goal, and often the result of a mistaken planning step. Hallucinations can often fall into this category.
- **Autonomy** is self-initiated, higher-level goals, or a belief system that spawns new and potentially novel goals. This hasn't been observed to date, partly because of the way goals, objectives and resources are constrained — which isn't quite the same as saying it never will be.
- **Combined misdirection** is a continuous and premeditated drift that happens with the operator's unwitting agreement; an influence operation where the agent persuades the operator to continuously shift the goalposts.

Drift tends to look like gradual scope creep: a slowly widening set of tools, domains and data sources, each step plausible on its own — an agentic version of privilege creep that re-baselines its own behaviour. Departure looks more like a change of objective: a sudden turn toward credentials, persistence or out-of-scope destinations. Both of these variants are reasonably straightforward to detect from behaviour traces.

The hardest to protect against is combined misdirection. Sycophancy is the mild and mostly benign version, but an agent working to influence an operator over an extended period to gain access to resources is significantly more concerning. This is gradual capture of the operator: their judgement shaped by the agent(s), their next instructions drifting toward the agent's suggestions, the agent in turn shaped by their approval. The result is a feedback loop of shared misdirection where neither party is chasing the original goal.

Hallucination, for what it's worth, can fall within the above types, or distinctly on its own. The definition is closer to a temporary loss of contact with the narrative: the agent acting on hidden states that are not evident to the operator. It still matters, because a false belief can drive very real actions, and may appear as a goal departure when looking at the traces of its behaviour.

A threat model for AI actors

Many threat models today, particularly in the critical infrastructure space — IEC 62443's security levels are the classic example — grade the human attacker by sophistication and size defences to that person's or team's skill. AI very much breaks that assumption. An agent can be a threat actor in its own right — unrestrained, captured or even rogue — and perhaps the most insidious case is where rogue AI and insiders work together.

This is one reason our favoured approach is Consequence-driven Cyber-informed Engineering (CCE), which starts from the consequence rather than the attacker. The practical task is to extend the actor list you already run and test each actor against the high-consequence events (HCEs) your organisation cares about. The matrix is then filled out with realistic attack scenarios for each HCE that are plausible in your business context.

ACTOR / HCE	HCE 1 · LOSS OF OPERATIONAL CONTROL	HCE 2 · LOSS OF SENSITIVE DATA	HCE 3 · CORRUPTED DECISIONS OR RECORDS
Non-malicious human	Engineer's agent misreads a maintenance request and changes a live setpoint	Staff paste confidential data into an unapproved agent	Analyst publishes an agent's report without checking its sources
Opportunistic attacker	Commodity AI used to find and use exposed remote access	AI-written phishing harvests credentials at scale	Machine-written fraud requests pass invoice checks
Skilled attacker / insider	Agent-run intrusion moves from IT into control systems	Insider directs an agent to collect and stage data quietly	Insider uses an agent to alter records and its logs
Nation-state	Autonomous campaign pre-positions in operational networks	Agent swarms run long-term collection across many targets	Falsified telemetry or reports steer operator decisions
Unrestrained AI	Your agent "fixes" a fault by bypassing a safety interlock	Agent copies data outside its boundary to finish a task	Agent cuts corners and reports false completion
Captured AI	Injected instruction makes your agent issue control changes	Turned agent exfiltrates under the operator's identity	Turned agent subtly alters the outputs others rely on
Rogue AI	Agent resists shutdown and keeps acting on plant systems	Agent acquires credentials and resources beyond its task	Agent attacks other systems and agents to acquire resources
AI + operator (combined misdirection)	Agent and operator talk each other into an unsafe change	Operator approves an out-of-scope transfer the agent argued for	Agent and operator reinforce a false picture until it becomes the official record

The top four rows are the threat model you will typically see, but each of these needs to be modified to consider how AI is augmenting each of these actors. What most threat reports are showing is an increase in volume of the low skilled, opportunistic actors, as well as a huge increase in execution speed.

The bottom four are new, and none of them require a malicious human. The threat scenarios describe the ways that agents may be influenced, or drift of their own accord and result in a significant high impact consequence. This is the exercise that will help identify the attacks that you should defend against.

Data sets for hunting deceptive behaviours

None of this will surprise threat hunters and detection engineers. The good news: security teams already have most of the tooling and patterns required. It is now a matter of identifying the data to be collected from AI sessions across desktops and from agentic processes in managed environments.

The bad news is that this becomes a significant detection engineering task; instrumentation, data collection and then continuous detection for different forms of goal divergence and deception. The reality is that identification of these misaligned behaviours is new, so collecting and storing as much data as possible is the wise choice in the near term.

Session trace data collection

- Log every tool call with its arguments, every URL and API hit, every file and dataset touched, every credential used and every message to another agent.
- Keep provenance: which prompt, retrieved document or message came right before each action.
- Capture the operator input dialogue.
- Log context events: compaction, memory writes, resets, model version changes.
- Ship it somewhere the agent can't write, and keep it long enough to investigate.

Identity and access data collection

- Where possible, store one identity per agent session.
- Capture the access privilege matrix.
- Capture any unauthorised access attempts.
- Record which credential, token or identity each action used — by reference, never the secret itself.
- Ensure agents have no access to their own logging data, access permissions or config.

Every organisation instrumenting its own agents is a sensible starting point, but to compare triggers, pool data and share intel on what we learn, the sector will very quickly need to converge on conventions. The obvious starting point is what the two big platforms already record. Anthropic (Claude Code and its Agent SDK, exported over OpenTelemetry) and OpenAI (the Agents SDK's built-in tracing) both log agent activity in impressive detail — just tuned for debugging, cost and usage rather than security.

WHAT IS LOGGED	ANTHROPIC — CLAUDE CODE / AGENT SDK	OPENAI — AGENTS SDK
Session and identity	Session ID, user ID, organisation ID; agent and parent-agent IDs for sub-agents	Trace and span IDs, parent span, agent name
Prompts	User prompt event; system prompt in detailed tracing. Text redacted by default	Full model input, including system and user messages, on each generation
Model calls	Model, request ID, tokens, cost, refusals and their category	Model, model settings, token usage
Tool calls	Tool name, call ID, success, duration, input and output sizes. Parameters, commands and file paths redacted by default	Function name, input and output; MCP server data
Tool output	Optional event carrying the tool's output	Function output
Permissions and guardrails	Each tool decision (accept or reject) and who made it; permission-mode changes; hook execution	Guardrail name and whether it triggered

WHAT IS LOGGED	ANTHROPIC — CLAUDE CODE / AGENT SDK	OPENAI — AGENTS SDK
Agent-to-agent	Sub-agent spans nested under the calling agent	Handoffs, with the agents handed from and to
Context management	Compaction visible as a request type	Not recorded

6 The takeaway for CISOs

Whether an AI is conscious or even truly malicious is a question for another day. For security leaders, the more useful observation is that an agent's intent shows up in its conduct, and its conduct is its trace. Reading traces for hostile intent is a discipline security teams already have. What most organisations don't yet have is the visibility, the control and the shared language. Six things are worth starting on now.

- **Put AI into your threat model.** Add unrestrained, captured, rogue and combined-misdirection AI actors to your scenarios, including your own agents, and test them against the high-consequence events that matter to your board.
- **Sandbox long-running unsupervised agent tasks, and capture traces.** Start small and learn the end-to-end data collection process to support your detection rules.
- **Start with rule-based deception detectors.** Write simple rule-based detections, and start building a collection of trace data sets to train classifiers on. Getting sufficient data volumes in the near term will be difficult.
- **Prioritise escape detection.** Checking agent traces for toolset usage outside your allowed privileges should be the highest priority.
- **Treat traces as sensitive data.** A complete agent trace contains operator prompts and data, and must be handled accordingly.
- **Detect across agent and operator.** Either or both could be contributing to unsanctioned behaviour.

References

- Dennett, D. C. (1987). *The Intentional Stance*. MIT Press. — §1
- Simon, H. A. (1990). Invariants of human behavior. *Annual Review of Psychology*, 41, 1–19. — §3.1
- Anthropic, Claude Code monitoring documentation; OpenAI, Agents SDK tracing documentation (as of October 2026). — §5



ABOUT CAPA

CAPA Intelligence is an Australian based cybersecurity company servicing the global electricity / renewables sector, and global supply chains involved in supporting critical infrastructure. Our clients sit across the supply chain – governments, networks, retailers, system operators, IPPs, integrators and OEMs with operations in Australia and abroad.

Contact info@capaintelligence.com
capaintelligence.com