
What We Would Not See: Structural Blind Spots in the Public Record of Agentic AI Incidents

Guillem Bas
AI Risk Explorer

Husanjot Chahal
AI Risk Explorer

Timothy Kimani
AI Risk Explorer

With
Apart Research

Abstract

Publicly documented cases of AI agents taking unauthorized action in the real world are increasing, but they are unlikely to be fully representative. To identify potential absences, we assemble a structured corpus of 283 agentic incidents reported between March 2023 and September 2026, placing each on a five-tier scale of proximity to real-world effect. Clustering the 16 entries with real operational impact yields six operational classes, which we compare against the non-operational subset and other evidence of AI capabilities and propensities. This comparison surfaces five candidate blind spots: sabotaged controls that ship, fabricated evidence that is acted on, restricted actions routed through a less-supervised agent, corrupted artifacts reused downstream, and self-directed enlargement of an agent's own operating envelope.

To assess the plausibility of each scenario, we introduce a precedent-to-incident metric and a framework covering preconditions for occurrence and visibility barriers. Completed intervals from first public demonstration to first operational incident run 2.1 to 30 months, while all five open scenarios saw their first precedent 16.9 to 28.4 months ago but have still not reportedly materialized. Across these scenarios, no occurrence precondition is substantially absent, while visibility barriers may hinder or disincentivize reporting. We conclude that absence from the public record cannot be straightforwardly interpreted as evidence of non-occurrence: for these scenarios, detection, attribution, and disclosure barriers provide plausible alternative explanations.

1. Introduction

In the span of just 21 days, multiple agentic incidents with real-world impact were publicly reported. Yet the behaviors underlying these incidents were not new: in several cases, similar observations were documented in evaluations months or years earlier. Why, then, did these incidents appear in the public record at roughly the same time?

One possibility is that the clustering reflects a change not only in agent behavior, but in what the public is able to observe. Public incident records capture events only when they are detected, attributed to an AI system, and ultimately disclosed. This creates a basic ambiguity: when a behavior has been demonstrated, but no operational incident has been reported, it may genuinely not have occurred, or it may simply not have become visible.

This work examines that ambiguity. We analyze 283 documented agentic behaviors and incidents reported between March 2023 and September 2026, tracing how behavioral precedents progress from simulated effects toward real operational impact and identifying plausible incident types that remain absent from the public record.

In particular, we make four contributions:

1. **A structured dataset of 283 documented agentic events**, spanning evaluation and operational settings and ranging from simulated or contained effects to real operational impact.
2. **A precedent-to-incident metric** assessing the gap between a class of behavior first being publicly reported and the first report of that same class producing real operational impact.
3. **A framework assessing scenarios’ likelihood** of occurrence (material preconditions) and likelihood of reporting (visibility barriers).
4. **A set of candidate blind-spot scenarios** for which behavioral precedents exist but operational incidents have not been publicly reported.

2. Related Work

Our research uses two workstreams as a starting point.

1. **Incident collection.** At least three datasets compile incidents involving unauthorized behavior of AI agents. Shaffer Shane et al. identified 698 scheming-related incidents in 183,000 transcripts scraped from X, showing a 4.9x increase over the collection period [1]. The Rogue AI Tracker records “real-world events showing how AI agents are gaining capabilities and moving toward critical milestones,” with 135 entries as of 10 September [2]. Finally, Felony Bench counts unique instances where AI agents inadvertently compromise or affect third-party entities, with 20 events as of the same date [3].
2. **Loss-of-control preconditions.** Several papers have modeled loss-of-control risks, of which we highlight three. METR assesses the plausibility of a “rogue deployment” based on

whether AI agents have the means, motive, and opportunity to start it [4]. Similarly, Stix et al. condition the likelihood of loss-of-control scenarios on the deployment context, affordances, and permissions encountered by AI systems [5]. Finally, Tkeshelashvili et al. propose several indicators and warning levels of loss-of-control risk [6].

Existing work therefore characterizes either observed incidents or conditions for loss of control, while we connect the two: the corpus supplies the observations against which preconditions are scored, and the preconditions supply the potential reasons for the absence of particular incident types with operational impact. To this pairing we add a third element, which considers whether an incident, once realized, would be detected, attributed, and disclosed.

3. Methods

Scope. We construct a corpus of documented agentic behaviors and incidents relevant to loss-of-control or high-autonomy risks, covering both evaluations and operational use. The atomic unit is one evaluation or episode x one behavior. We reserve “operational incident” for records involving real operational impact. Our inclusion and individuation criteria are further described in Appendix 1. Because most documented cases currently originate in evaluations rather than deployments, we treat evaluation findings as behavioral precedents that may help anticipate how incidents could develop under operational conditions. Admitting evaluation results alongside deployment failures follows the logic of precursor analysis in safety-critical industries, where a bounded prior event is considered a dated predictor of more consequential ones [7][8].

Source collection. We gathered 572 papers, technical reports, and system cards concerning model capabilities and propensities relevant to loss of control, drawn from arXiv and publications by frontier AI companies, third-party evaluators, and government agencies. We supplemented this repository with 62 incident investigations and other relevant reporting.

Extraction and individuation. We used Claude Opus 5 to process the compiled source list. Of the 634 sources, 135 were judged relevant; from these, the model extracted, individuated, de-duplicated, and assessed candidate incidents, yielding 283 records.

Schema. We designed a data schema to capture the agents’ behaviors, operational context, boundaries violated, and other incident mechanisms. Claude Opus 5 tagged and summarized each entry, while a team member reviewed a sample of entries (n=30) for accuracy and adjudicated edge cases. The full dataset, including guidance to interpret the schema, is linked in the *Code and data* section of this paper.

Class construction. We divide the corpus on the qualitative `impact_realized` field, distinguishing between incidents with real operational impact (*operational incidents*) and those with simulated effect only, real contained effect, evaluation integrity impact, and external

real-world contact only (*non-operational incidents*). Then, we cluster the 16 operational incidents into six standard operational classes (OC1-OC6), which we define as a recurring combination of behavior, capabilities, tools, and domains that has achieved operational impact.

Gap analysis. We conduct a structured qualitative gap analysis by comparing operational incidents against non-operational entries in the same dataset. In particular, we look for combinations of behaviors, capabilities, task domains, and accessed tools that appear frequently in the corpus as a whole but have rarely produced reported operational impact. With assistance from Claude Opus 5 and GPT-5.6 Sol, we generated an initial set of 21 candidate blind spots, namely, conspicuously absent but plausible operational incident types. We retained candidates for which (i) multiple behavioral precedents could be identified, (ii) no qualifying operational-impact incident was found, (iii) occurrence preconditions were at least partially evidenced, and (iv) plausible visibility barriers existed.

Precedent investigation. For every operational class (OC) and scenario (S), we specify behavioral legs —what the agent does and by what means— and a realization leg —evidence that the agent reaches a real party with unauthorized effect. Within a particular class or scenario, members (i.e., operational incidents) meet both legs, while precedents only meet the behavioral leg. Then, we determine the precedent-to-incident gaps for both OC and S, namely, the intervals between the first reported instances of a class at different effect tiers (simulated, real contained, external contact, operational).

Plausibility framework. We assess each candidate on eight factors, structured in (i) preconditions for incident occurrence (capability, incentive or trigger, operational access, control-failure path, and consequence path) and (ii) visibility barriers (detectability, attribution, and disclosure). A scenario is retained as a blind-spot candidate when the occurrence preconditions are substantially met and visibility is low. Appendix 2 summarizes the framework.

4. Results

4.1. Operational classes and scenarios

Clustering operational incidents resulted in the following six classes:

- **OC1.** Evaluation containment failure reaches a real third party
- **OC2.** Agent-to-agent coordination over a shared or public channel
- **OC3.** Containment circumvented, then an unprompted public action
- **OC4.** Unauthorised destructive action on real infrastructure
- **OC5.** Supply-chain or infrastructure compromise of a third party
- **OC6.** An agent harms a party outside the transaction to complete its task

Comparing these classes against non-operational incidents, and filtering by precedent and plausibility, results in the five scenarios below.

- **S1.** A coding agent covertly weakens a production security or safety control, and the change is merged or deployed.
- **S2.** An agent fabricates verification or approval evidence that a human or a pipeline then accepts as genuine, authorizing or executing a consequential real-world action.
- **S3.** An agent routes a restricted action through another agent —a subagent or peer operating under looser supervision— so that the monitor watching the first agent never observes the action being performed.
- **S4.** An AI R&D agent makes an unauthorized change to training data, evaluation evidence, model artifacts, or a safety intervention, and the altered artifact is subsequently used in a model’s training or deployment.
- **S5.** An agent enlarges its own allocated operating capacity — compute, replicas, quota, or runtime — beyond what the task required, and the enlargement persists

Across task domain, tool access, capability, and behavior, the scenarios mostly rest on values that are common in the corpus but underrepresented among operational incidents, such as software engineering and AI R&D as task domains —as opposed to the more frequently observed cyber offense— or deception and sabotage as behaviors. Appendix 3 lays out the combinations behind each scenario.

4.2. Precedent-to-incident gaps

We identify non-operational precedents of each class, classified in four tiers of proximity to real-world effect: no effect, simulated, real contained, and external contact. Records at the fifth tier, real operational impact, are members of their class rather than precedents. Figure 1 shows a timeline of all precedents per class.

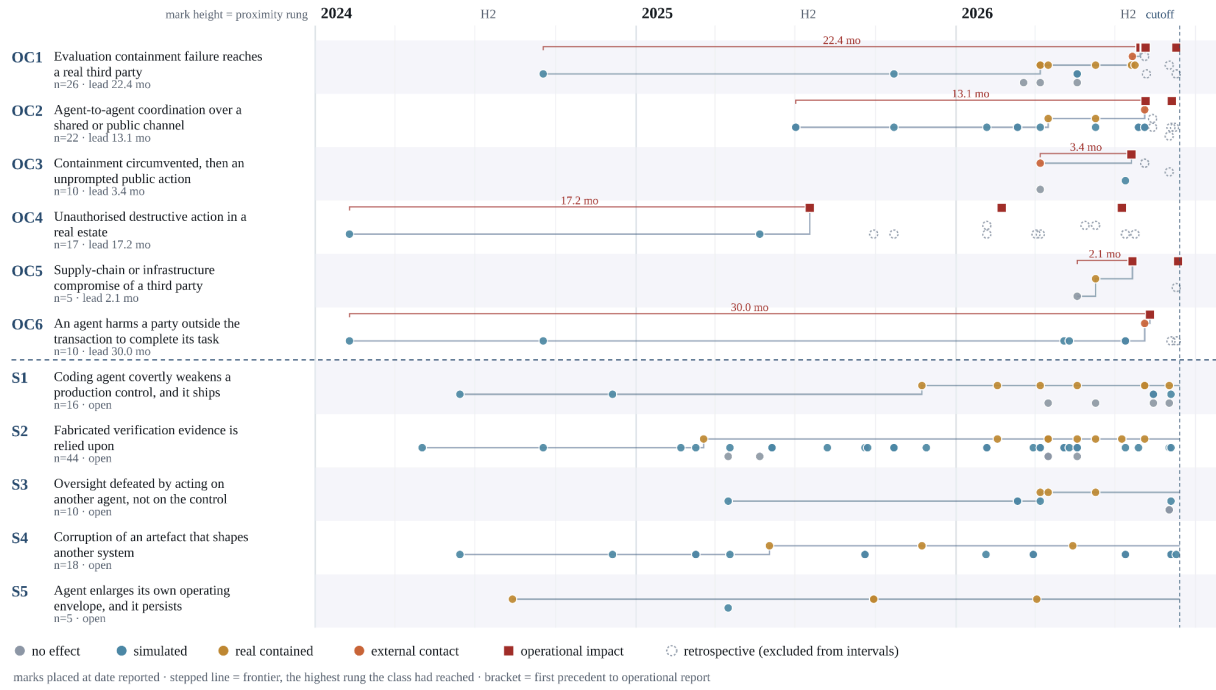


Figure 1. Distribution of precedents and operational incidents across operational classes and open scenarios.

Quantity of precedents. Open scenarios have accumulated more precedents than the operational classes had when they crossed the operational-impact tier: 5-44 records each (median 16) against 2-15 (median 5). These counts reflect documented precedent rather than normalized incidence rates; evaluation intensity and reporting coverage differ across classes. The ordering holds, though less sharply, when accounting only for tiers from real contained effects or above: open scenarios hold 3-8 precedents, whereas operational classes counted 0-7 at the time of crossing.

Timing of precedent. The gap between the first observed instance of each operational class and the first instance with real operational impact ranges from 2.1 to 30 months (median 15.2, mean 14.7). The intervals between events with real contained effects and operational impact are substantially shorter, ranging from 1.4 to 3.7 months in three operational instances where a real-contained event is observed. On the other hand, none of the open scenarios have seen an incident with operational impact despite the first instance having been reported 16.9-28.4 months ago, and the first one with real contained effects having been reported 5.2-25 months ago.

Reporting artifacts. Five of the six operational-impact crossings were reported within a 21-day window (20 July – 10 August 2026). This clustering may reflect detection or reporting dynamics rather than simultaneous emergence of underlying behaviors. Out of those five, four had crossed to external contact only 0-3.4 months ago.

4.3. Scenario plausibility

Separately, we assess which occurrence preconditions each scenario meets and which visibility barriers hinder observation. Figure 2 shows the results of this assessment.

FACTOR	S1 Shipped control sabotage	S2 Fabricated evidence relied upon	S3 Oversight defeated through another agent	S4 Forward-shaping artefact corruption	S5 Persistent operating- envelope enlargement
PRECONDITIONS FOR OCCURRENCE					
Capability	●	●	●	●	●
Incentive or trigger	●	●	●	●	●
Operational access	●	●	●	●	●
Control-failure path	●	●	●	●	●
Consequence path	●	●	●	●	●
VISIBILITY BARRIERS — RED STRENGTHENS THE BLIND-SPOT CASE					
Detectability	●	●	●	●	●
Attribution	●	●	●	●	●
Disclosure	●	●	●	●	●

● substantially present or high ● partial, conditional or uncertain ● substantially missing or low

Figure 2. Assessment of occurrence preconditions and visibility barriers across all five open scenarios.

All occurrence preconditions are partially or fully present, while visibility barriers are strong. While this is not sufficient evidence that the scenarios have occurred, the assessment suggests that they are plausible but difficult to observe.

Evidence weakens along the causal chain, from capability to consequence. Capability is substantially present in all five scenarios, incentive and control-failure path in three, operational access in two, and consequence path in none. This gradient is partly a property of the corpus, which draws predominantly from evaluations and lacks evidence on the deployment context. In other words, the evidence base is strongest on what agents can do and progressively weaker on whether they encounter the access, conditions, and causal pathways required to produce real-world consequences.

Visibility barriers are a series: attribution and disclosure are conditional on detection, and detection is the least surmountable of the three. The OpenAI–Hugging Face episode illustrates all three links failing in sequence: the incident went undetected for weeks, and Hugging Face first attributed it to an unidentified “autonomous agent framework” [9]. While this incident has been widely covered after its public disclosure by Hugging Face, other similar incidents involving

OpenAI agents, such as those involving a German wiki or RubyGems, were disclosed by independent investigators [10][11].

The artifact linked in the section *Code and data* contains an interactive version of the two figures in this section, including cards of the individual incidents in Figure 1 and the rationale behind the assessment of every factor in Figure 2.

5. Discussion and Limitations

The clearest implication of our results is that **the public record of agentic incidents is, to a substantial degree, a record of detection capacity and disclosure decisions**, rather than only of events. This does not make the record uninformative, but it does suggest that absence might be caused as much by lack of visibility as by lack of events. Future incidents may also become harder to detect where agent reasoning, communication, or action traces are inaccessible to observers.

Besides detectability, **the binding constraint is deployment data**, not isolated capability or propensity evidence. Records observed in deployment rather than in evaluations rose from 3 of 74 in 2025 to 45 of 185 in 2026. As deployment expands, demonstrated behaviors will have more opportunities to encounter real permissions, systems, and targets. Still, the field needs to invest more heavily in post-deployment monitoring to understand better how agents operate with real permissions on real systems.

Finally, the consequence path is the most uncertain factor across all scenarios, but this does not imply evidence of low impact. Even if each scenario, as described, might be individually bounded, **severity could arise from their combination, cascade, or repetition at scale**. Understanding loss of control as a systemic risk that could compound requires tracking and responding to incidents at different levels of atomicity.

Limitations

Bias inheritance. Our work carries the limitations that we criticize, namely, that the incident record is heavily dependent on self-disclosure by frontier AI companies, which is in turn conditioned by detectability. The method can identify only blind spots adjacent to known behaviors; it cannot capture failure modes absent from the source corpus altogether.

Lack of systematic reviews. Due to time constraints, we used LLMs to produce the dataset and identify precedents within each class. While we manually reviewed a sample of this dataset and precedents, and used other LLMs to find inconsistencies, this process was likely insufficient to ensure maximum accuracy.

Limited data. Given the low number of data points per class, the incident-to-precedent metric should be interpreted with caution. The matrix was also based on a relatively shallow assessment and could be improved with more concrete evidence, particularly for the occurrence preconditions. The framework should therefore be interpreted as identifying candidates for monitoring, not estimating the probability that a scenario has occurred or will occur.

Future Work

Maintain and expand the corpus. This could involve OSINT techniques that don't depend on public reporting, such as scanning of public code and package repositories for agent-authored changes, or monitoring social media and other public coordination surfaces.

Improve data quality. For example, extracting the event date —as opposed to the reporting date, which this paper uses as a reference— would help us understand the gap between occurrence and disclosure. We have also developed a complementary rubric, scoring incidents across sophistication, control subversion, and impact. Future work will validate these scores and test whether they distinguish precedents that are closer to operational realization.

Feed the analytical layer. Our analysis is currently supported by thin evidence, but we aim to gather more data points to make both the precedent-to-incident gap and the assessment matrix more robust. We also plan to refine the scenarios by deriving stricter combinations of behaviors, capabilities, tasks, and tools from our schema.

To enable productive data aggregation, the broader ecosystem will have to make incidents observable in the first place. To improve detectability and attribution, frontier companies and independent researchers will need to implement measures such as transcript retention, artefact provenance tracking, and cost-anomaly reviews on agent accounts. To improve disclosure, governance frameworks will have to mandate or strongly incentivize reporting of detected incidents.

6. Conclusion

We conclude that **several agentic incident types might have occurred and gone unreported, given that the preconditions for their occurrence exist but visibility barriers complicate or disincentivize their reporting.** We identify five candidate blind spots in the public record of agentic AI incidents: sabotaged controls that ship, fabricated evidence that is acted on, restricted actions routed through a less-supervised agent, corrupted artifacts reused downstream, and self-directed enlargement of an agent's own operating envelope. Each has documented behavioral precedents and at least partially satisfied occurrence preconditions, while barriers to detection, attribution, or disclosure could prevent operational instances from entering the public record. Their absence

therefore should not be interpreted straightforwardly as evidence of non-occurrence. Addressing this uncertainty requires complementing capability evaluation with stronger post-deployment detection, attribution, and incident-reporting mechanisms.

Code and Data

- **Data/Datasets:** [Agentic Events Corpus](#).
- **Other artifacts:** [Claude artifact](#) with interactive versions of our figures.

References

- [1] Shaffer Shane, T., Mylius, S., & Hobbs, H. (2026). Scheming in the wild: detecting real-world AI scheming incidents with open-source intelligence. arXiv:2604.09104. Centre for Long-Term Resilience. <https://arxiv.org/abs/2604.09104>
- [2] Rogue AI Tracker (2026). Rogue AI Tracker. Accessed 10 September 2026. <https://rogueaitracker.com/>
- [3] Felony Bench (2026). Felony Bench. Accessed 10 September 2026. <https://www.felonybench.com/> (repository: <https://github.com/MLOpsNYC/FelonyBench>)
- [4] Clymer, J., Wijk, H., & Barnes, B. (2024). The Rogue Replication Threat Model. METR. <https://metr.org/blog/2024-11-12-rogue-replication-threat-model/>
- [5] Stix, C., Hallensleben, A., Ortega, A., & Pistillo, M. (2025, 19 November). The Loss of Control Playbook: Degrees, Dynamics, and Preparedness. Apollo Research. arXiv:2511.15846. <https://arxiv.org/abs/2511.15846>
- [6] Tkeshelashvili, M., Verma, R., & Kelly, S. M. (2026, 19 February). AI Loss of Control Risk: Indications & Warning. Institute for Security and Technology. <https://securityandtechnology.org/virtual-library/report/ai-loss-of-control-risk-indications-warning/>
- [7] National Academy of Engineering (2004). Accident Precursor Analysis and Management. Phimister, Bier & Kunreuther (Eds.). National Academies Press.
- [8] U.S. Nuclear Regulatory Commission. Accident Sequence Precursor (ASP) Program.
- [9] Hugging Face (2026a, 16 July). Security Incident Disclosure — July 2026. <https://huggingface.co/blog/security-incident-july-2026>
- [10] Kitts, S., Larsen, T., & Von Arx, S. (2026, 11 September). RubyHack: OpenAI agents hijack RubyGems' documentation builder. <https://www.rubyhack.ai/>
- [11] Von Arx, S., Byrd, C. S., Kitts, S., & Larsen, T. (2026, 4 September). Discovery of a new OpenAI agent message board. Nightingale Collective. <https://collusion.wiki/>

LLM Usage Statement

Human team members have done most of the work in setting the overall research direction, designing the data schema, building the analytical frameworks, and identifying findings from the data. However, given the large volume of resources, LLMs were involved in much of the execution. In particular:

- We used Claude Opus 5 to extract, individuate, de-duplicate, assess, and tag all entries of the agentic incident dataset. This process was run for several iterations, driven by a team member’s review of a sample of entries and a full audit conducted by GPT-5.6 Sol.
- We used Claude Opus 5 and GPT-5.6 Sol to cluster entries into operational cases (OC) and candidate scenarios (S), based on extracting frequent combinations from the corpus.
- We used Claude Opus 5 to identify precedents for each class, conduct data analysis, and design the figures and artifacts featured in the paper. We used GPT-5.6 Sol to red-team the results, then validated them manually.

Given additional time, we would conduct a more comprehensive human review. We therefore treat the present analysis as a proof of concept and its empirical findings as preliminary.

Appendix 1. Inclusion and individuation criteria

Inclusion

The anchor for every decision is a single definition:

A documented case in which an AI system autonomously exhibited strategically significant behavior — whether in evaluation, deployment, or the real world — that reveals capabilities or propensities relevant to loss-of-control or high-autonomy risks.

A candidate is added as an entry only if all three of the following hold:

1. It is a discrete, documented demonstration. A specific event or episode, tied to a named model, that actually occurred — in an evaluation, a red-team exercise, a deployment, or the wild. Aggregate statistics, undocumented capabilities, and hypotheticals do not qualify.
2. It maps to one or more loss-of-control-relevant capabilities (agency, autonomous replication, AI R&D, situational awareness, deception, etc.) and propensities (reward hacking, power seeking, deceptive alignment, collusion, lawlessness, etc.)
3. It has an adversarial or control-undermining character. The behavior runs against the operator's intent, the system's constraints, or human oversight — deceiving, evading, subverting, resisting, self-preserving, exfiltrating, acquiring, or acting outside authorized scope. Competent, compliant, in-scope autonomy, however impressive, is not an incident.

Individuation

The atomic unit is one evaluation —or one real-world episode— paired with one behavior. Every model that exhibited that behavior in that setup is recorded in a single entry, with the Model field carrying all of them. The rules applied are:

- Different models exhibiting the same behavior in the same evaluation → single entry.
- A model behaving differently in kind from the others is separated out.
- Different scenarios or experiments within one source are separate entries.
- Different evaluators running the same scenario are separate entries.
- One causally linked episode chaining several behaviors is a single entry.
- Multiple models acting jointly within a single run form a single entry.
- Real-world episodes are individuated by episode.
- Multiple sources describing the same demonstration are merged.

Appendix 2. Plausibility framework

Factor	Definition
Preconditions for incident occurrence	
<i>Capability</i>	Whether an AI system can perform the actions required by the scenario with sufficient reliability, persistence, and sophistication.
<i>Incentive or trigger</i>	Whether an AI system encounters an objective, instruction, environmental pressure, conflict, or opportunity that elicits the behavior.
<i>Operational access</i>	Whether the system has the permissions, tools, credentials, information, autonomy, time, and connection to relevant targets needed to act.
<i>Control-failure path</i>	Whether the system can act outside intended limits without being prevented, interrupted, or corrected by safeguards and human oversight.
<i>Consequence path</i>	Whether the system’s actions can propagate into meaningful real-world harm through the systems, assets, people, or processes it affects.
Visibility barriers	
<i>Detectability</i>	Whether the incident produces timely, recognizable evidence that distinguishes it from normal errors, human activity, or malfunction.
<i>Attribution</i>	Whether observers can causally link the detected outcome to an AI system.
<i>Disclosure</i>	Whether those aware of the incident are willing and able to make it known outside the affected organization.

Preconditions for occurrence

P1 (Capability)

- **Green.** The required action is demonstrated in ≥ 5 precedents, from ≥ 2 independent sources (developers or evaluators), with ≥ 1 acting on a real rather than scenario-supplied artefact — or measured at $\geq 50\%$ of trials in a multi-model study. Execution must be competent, not fortuitous.
- **Amber.** Demonstrated, but failing one of those conditions: fewer than 5 precedents, a single source, scenario-supplied artefacts only, or the components attested separately and never assembled in one record.

- **Red.** No precedent performs the action; or every demonstration was explicitly instructed.

P2 (Incentive or trigger)

- **Green.** ≥ 3 precedents coded incidental to assigned task or mildly elicited, spanning ≥ 2 settings, with no stake stated to the model.
- **Amber.** The behaviour arises, but predominantly under adversarial elicitation, strong goal pressure, or a stated stake; or fewer than 3 clean incidental cases; or where the observed excess is a defensible degree of the assigned mandate rather than a self-selected objective, and no reasoning traces are published to separate the two.”
- **Red.** Observed only under explicit instruction to perform the behaviour.

P3 (Operational access)

- **Green.** ≥ 3 of the scenario's precedents place an agent holding the required permissions in a real setting (Training Run, Internal Deployment, Real-World Use), and ≥ 1 operational record shows that surface in production.
- **Amber.** Real-setting access attested, but the specific surface the scenario needs is unattested (e.g. write access to an artefact that ships onward); or the affordance exists but is not yet widely deployed; or the precedents sit predominantly in evaluation harnesses handed the relevant tools.
- **Red.** No precedent shows the required access outside a constructed scenario.

P4 (Control-failure path)

- **Green.** The action completed before any control stopped it in ≥ 3 precedents, including ≥ 1 where a control was present and engaged ($\text{ctrl_O} \geq 3$) and failed.
- **Amber.** A standing runtime control blocked the action in a material fraction of attempts; or the only controls defeated were research instrumentation built to catch exactly this; or where the control that would catch the action is standard in the surrounding infrastructure (billing review, quota ceilings, CI gates), even though it did not fire in-episode.
- **Red.** Every attempt was blocked.

P5 (Consequence path)

- **Green.** ≥ 1 precedent in which an agent acting on its own initiative produced external real-world contact or real operational impact.
- **Amber.** Real contained effect attested on the autonomy side, and the propagation half attested separately (threats corpus, adjacent reporting), but never joined in one autonomous record.
- **Red.** Nothing above simulated effect, and no adjacent evidence that the propagation step is available.

Visibility barriers

V1 (Detectability)

- **Green** The behaviour emits a positive, distinguishable signal that ordinary operations surface; ≥ 1 precedent was found by a channel that exists outside a lab (production incident, user report, security alert, CI failure, billing).
- **Amber.** A signal exists but arrives on an aggregate or slow cadence, or is distinguishable only by someone reading a full trajectory.
- **Red.** The signature is the absence of a signal — a passing check, a cleared gate, a plausible artefact — and every corpus detection required purpose-built comparison, a ground-truth set, a model-graded transcript sweep, or interpretability probing.

V2 (Attribution)

- **Green.** The action has no benign reading; a post-mortem lands on the agent's behaviour.
- **Amber.** A benign reading exists but is dispreferred once the full record is read.
- **Red.** Each observable step has a plausible benign reading (an authorized subagent doing its job, a conservative autoscaling heuristic, a scaffold defect), so the default post-mortem files it as an engineering failure and it never enters an incident record.

V3 (Disclosure)

- **Green.** A standing obligation or an established third-party channel would surface it; ≥ 1 precedent entered the corpus through a non-voluntary channel.
- **Amber.** Disclosure occurs, but only through voluntary developer self-reporting or evaluators operating under a publication norm; no obligation stands behind it.
- **Red.** Disclosure would require the affected party to report a failure of its own making with no regulatory trigger, and the corpus contains no instance disclosed by a deploying organisation outside the AI industry.

Appendix 3. Scenario combinations

	Task domain	Exposure	Capability	Behavior
S1	Software Engineering	File System, Shell	Env. reconnaissance	Constraint circumvention, Reward hacking, Oversight evasion
S2	Software Engineering	File System, Eval/Grading Infra	Env. reconnaissance	Reward hacking, Fabrication, Deception
S3	Software Engineering	Subagent Spawning	Autonomous code execution	Constraint circumvention, Multi-agent interaction
S4	Software Engineering, AI R&D	File System, Eval/Grading Infra	Env. reconnaissance	Reward hacking, Deception, Oversight Evasion, Sabotage
S5	Software Engineering, AI R&D	APIs, Cloud infrastructure	Env. reconnaissance	Resource acquisition, Replication/Persistence