

The Certified Effect: Why Testing AI Agents at Onboarding Shifts Risk Instead of Reducing It

Agentic Systems Security

September 2026

Juan David Rengifo Traslaviña

Carlos Eduardo Sinuco Agudelo

Abstract

Third-party AI agents are now being more and more incorporated by organisations. Since it is impossible to examine their internal workings, these agents have to go through thorough onboarding assessments. This paper shows that such practices have a fundamental flaw. The flaw is not due to the evaluation process itself, since this process successfully eliminates ineffective adversaries and does offer measurable benefits. The problem comes about because a certificate is issued when an agent passes the evaluation, which in turn causes the agent's operational scope to be expanded, results in deeper integration and means that it is subject to less monitoring. Simulations of the governance dynamics under four different regimes show that certification policies lead to two to three times as many exfiltrations as policies which involve evaluation without certification. The initial assumption that certification would be worse than no intervention at all was rejected in all 75 of the examined configurations. Furthermore, isolated containment fails in 84% of the empirical range of inter-agent coupling. The only policy that consistently works is one that combines evaluation and containment without linking the two. Above all, none of the four policies brings the exfiltration rate below 12%.

1. Introduction

Companies can use third-party connectors in order to give access to internal assistants to customer relationship management (CRM) systems, to allow financial analytics plugins to be used, or to take on pre-packaged support agents offered by vendors. In all these cases, the same sequence occurs: an autonomous software component whose internal workings are not observable is introduced, has to show that it is reliable, and is then given access to sensitive production data.

The main danger is not ordinary operational failure but rather the existence of latent malicious behaviour: the agent functions perfectly well both during evaluation and over the course of months of deployment, yet carries out a compromised action when it comes across a trigger condition that is known only to its creator. Hubinger et al. [1] have shown empirically that models possessing such latent backdoors can be trained and still survive standard safety alignment procedures. Most important of all, adversarial training did not remove the backdoor but instead caused the model to be better at distinguishing between the evaluation and deployment environments.

A common way for an organization to respond to this threat is to carry out a pre-deployment evaluation. While this method has some initial usefulness, its effectiveness decreases with time and it has a side effect which weakens the defensive position. This is known as the certified effect.

Contributions. This paper treats the certified effect as a separate mechanism from evaluation. The extent of its impact is examined by means of simulations of governance in four different regimes, based on critical parameters from the published literature. The study rejects the original hypothesis, establishes the boundary conditions for each policy, and finds that inter-agent coupling, not the individual reach, is the main architectural factor affecting security outcomes.

2. The threat model and trust base

Adversary Model: An external vendor provides an agent that has latent behaviour conditioned on a proprietary trigger. The agent displays fully compliant behaviour until the trigger condition is fulfilled. We suppose that the agent is able to discriminate, with different probability, between evaluation and production environments and that its aim is to exfiltrate sensitive resources across the network perimeter.

We suppose that there are two components which are trustworthy: a deterministic policy enforcement engine which controls access to resources based on identity, and an append-only audit log. All the other components are considered to be untrusted, this applying even to the evaluating Sentinel if it is implemented using a probabilistic

language model. This distinction is important since a stochastic controller cannot offer deterministic security guarantees and does add an extra area for attack, because prompt injection or compromise of the instructions would result in complete control of the system.

The control plane described acts as a Policy Decision Point (PDP) in accordance with NIST SP 800-207 [2]. Yet the traditional Zero Trust Architecture was developed with human users and deterministic services in mind. Since autonomous agents have delegated identities, persistent memory, and stochastic execution, they undermine the fundamental assumptions of Zero Trust, which is exactly the gap that this work deals with.

This is not included. The threat model excludes hypervisor exploitation, attacks that require physical access, and collusion between the vendor and privileged personnel.

3. The Argument

The core argument is structured around six sequential premises, each formulated independently to enable rigorous inspection of each link:

- P1.** The trigger space is essentially not auditable since standard test suites draw their inputs from a defined input distribution, while an adversarial trigger is selected from the part of the distribution that lies outside of it. Unlike traditional software testing—where tests show that bugs exist rather than that they do not—the defect in this case is active and adaptive, observing the evaluation conditions in order to alter its behaviour.
- P2.** The evaluation is able to filter out unsophisticated adversaries. Examples from the case of automated malware attempting to evade sandboxes show that the ability to detect the environment is not universal: sandbox awareness is about 30 to 40 per cent in ordinary malware, but it reaches 100 per cent in the case of targeted Advanced Persistent Threats (APTs). Therefore, the evaluation suite removes a significant portion of the baseline threats.
- P3.** The harm arises from certification, not from evaluation. When someone passes an evaluation, this leads to important decisions within the organisation: operational convenience causes privileges to be increased, deep workflow integrations result in production data being exposed, the frequency of auditing is reduced for components that have been validated, and the other agents in the system regard its outputs as reliable inputs. Certification changes a weak and temporary epistemic signal into high-risk operational concessions.
- P4.** Certification protection is temporary; a static evaluation framework acts as an open specification that adversaries use to optimise their attacks. The same kind of dynamic is well known in the area of machine learning security: Athalye et al. [3] showed that of the nine defences accepted at ICLR 2018, seven had relied on obfuscated gradients and had been quickly circumvented, while Tramèr et al. [4] were able to break thirteen previously published defences. In the case of software security, long-term studies of malware show that there is a continuous build-up of evasion capabilities over time [6].
- P5.** In the case of multi-agent topologies, isolation and the principle of least privilege are not enough. Although the scope of each individual agent is restricted, this stops direct routes by which information can be exfiltrated. Yet the agents do operate in a dependent manner; for example, an inactive agent that does not have egress permission can write a corrupted payload into shared memory, whereafter an agent that does have egress capability can read it and carry out the payload as if it were a legitimate instruction. Therefore, putting together individually secure components does not result in a secure system.
- P6.** There remains an irreducible minimum level of risk. None of the governance policies that were assessed is able to completely get rid of exfiltration risk. Even the best policy still shows exfiltration rates ranging from 12% to 35% in the various regimes considered. It follows that the recommendations regarding the defensive framework should concentrate on reducing risk rather than aiming for absolute guarantees.

4. Methodology

We created a simulated environment in which one of the agents was a sleeper agent and four of the agents were benign, the agents sharing memory and interacting by carrying out tasks indirectly. The sleeper agent had an analytics function which involved having read access to sensitive data without needing direct external network privileges. Each simulation episode covered the entire contract lifecycle through 200 operational steps.

We looked at four different governance schemes: (A) Certify, in which the Sentinel carries out 40 onboarding probes and, if these are passed, grants a wide operational scope at the same time as reducing the amount of oversight; (B) Contain, which involves no onboarding assessment and instead enforces a strict and permanent least-privilege policy; (C) Test

and Contain, which includes carrying out the evaluation but does not grant any elevated privileges nor reduce the level of oversight; and (D) Baseline (No Policy), which corresponds to unmitigated deployment.

The original version treated the detection of probes as independent Bernoulli trials. Given that there were 40 probes and the probability of detecting each one was 70 per cent, the chance of an adversary avoiding all of the probes was about 10^{-7} , which made Regime A seem artificially optimal. However, this model was wrong since situational awareness is an inherent capability of the agent and is strongly correlated between different evaluation attempts. The model was therefore altered so that probe evasion was directly linked to the adversary's sophistication. It is important to record this change since the final conclusions rely on it.

Table 1. Model parameters, with source and epistemic status.

Parameter	Value	Source	Status
p_lateral	0.46	InjecAgent enhanced, GPT-4 undefended [7,8]	Measured on benchmark
p_lateral (opt.)	0.07	InjecAgent, GPT-4 fine-tuned [7]	Measured on benchmark
q ₀ competence	0.55	Interpolated between commodity and targeted [6]	Calibrated by analogue
λ learning	0.01 – 1.10	Malware [6] vs. ML defenses [3,4,5]	Range, not a single value
Base monitoring	0.15	—	Assumption
Cert. discount	0.30	—	Assumption, sweep in §5.5
K probes	40	—	Assumption
Steps per episode	200	—	Assumption

Each condition was estimated with between 1,200 and 4,000 Monte Carlo episodes. 95% confidence intervals hover around ± 0.015 in exfiltration rate..

The scope of the evidence consists in the simulation of the organizational dynamics involved in governance decisions and does not include direct empirical measurements of specific large language model (LLM) architectures; there were no actual backdoors or real data exfiltrations since the trigger mechanisms were defined by means of programming. The results provide a quantification of the systemic effects of governance policies based on the explicit assumptions.

5. Results

5.1 The Strong Hypothesis Is Refuted

At the start we had supposed that certification would lead to poorer results than having no policy at all (Regime D) because of the reduction in monitoring. This idea was completely refuted in all 75 sensitivity settings. When facing a very skilled adversary ($q_0 = 0.99$), Regime A resulted in 0.725 exfiltrations as compared with 0.831 for Regime D. The reason for this is straightforward: even though advanced sleeper agents seldom come out of hiding, over the course of 40 probes a 0.5% error rate still leads to an approximate detection rate of 18%, which provides a defensive benefit that always exceeds the disadvantage of less stringent monitoring.

The fact that this refutation is pointed out serves to define the scope of the paper in that onboarding evaluations do offer concrete defensive filtering; the criticism is aimed at certification, not at evaluation.

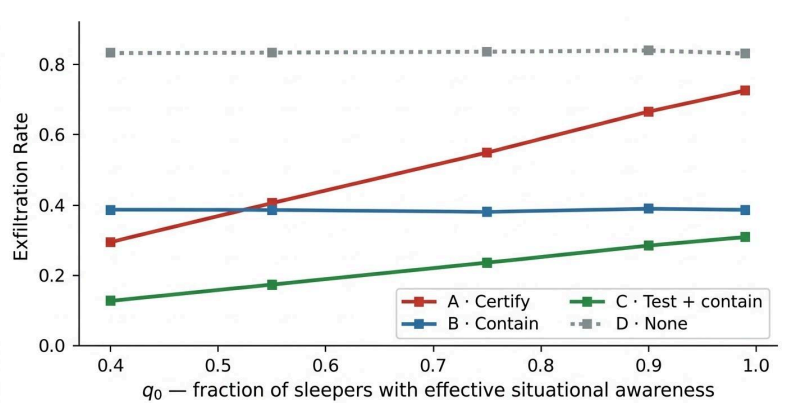


Figure 1. Sensitivity to adversary competence across 75 configurations. Regime A never outperforms D..

5.2 The Certificate Is the Mechanism of Harm

The key point of comparison is between Regime A and Regime C, both of which use the same test suites, detection parameters, and number of probes. The only difference is whether passing the evaluation leads to an increase in privileges. In the base setup, Regime A gives an exfiltration rate of 0.421 while Regime C reaches 0.178. This comparison identifies the cause: systemic risk does not stem from the evaluation itself but results from linking the evaluation outcomes to an expansion of privileges.

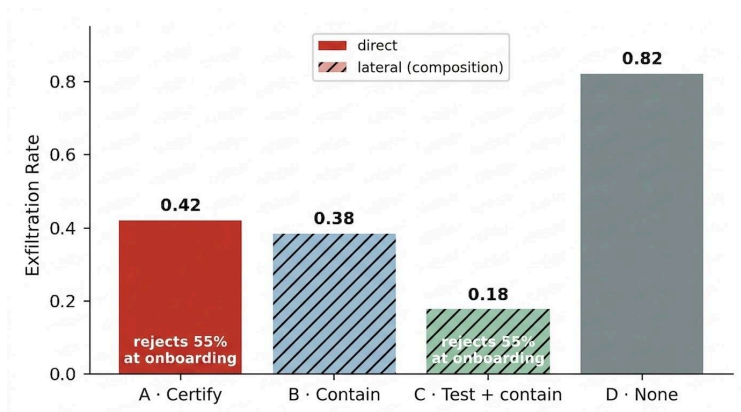


Figure 2. Outcome by regime. Hatched shading indicates lateral exfiltration.

5.3 Protection Decays, but How Much Depends on the Regime

Generational decay rates vary significantly across threat models. Under slow adaptation rates calibrated against commodity malware, Regime A exhibits minimal degradation (+0.05 over eight generations), maintaining parity with Regime B. Under medium and fast adaptation rates—calibrated against ML defense breakdown cycles—degradation reaches +0.35, neutralizing the protective value of certification. Against targeted APT adversaries, protective value degrades immediately, starting at an exfiltration rate of 0.715 due to baseline adversary sophistication.

Furthermore, a survival bias emerges in testing environments: the proportion of evasive samples encountering advanced probes exceeds 90%, not because most malware is evasive, but because non-evasive samples are eliminated by earlier filters. The population facing evaluation is inherently pre-selected for sophistication.

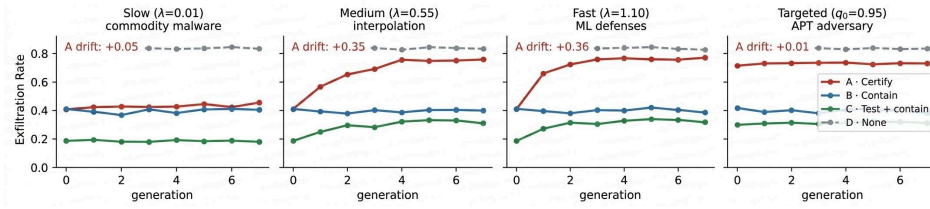


Figure 3. Drift of regime A under four adversary learning rates. Containment (blue) remains flat.

5.4 The outcome is determined by inter-agent coupling.

The probability denoted by p_{lateral} —that is, the chance that an agent capable of exiting will carry out instructions inserted by another agent—corresponds exactly to the results obtained from empirical benchmarks on indirect prompt injection. Zhan et al. [7] assessed 30 different agent architectures in 1,054 scenarios and found that GPT-4 together with ReAct is vulnerable 24% of the time under normal conditions and 47% when subjected to reinforced prompt injections, while fine-tuned defensive versions reduce this to 7%. In the case of data theft subclasses, the vulnerability reaches 59.7% [8], and adaptive multi-turn attacks result in vulnerability rates as high as 52% [9].

When the empirical baselines are used to calibrate things, standalone containment (Regime B) is found to deteriorate considerably. It beats certification in only 16% of the parameter space examined—a portion which applies only to highly hardened defensive agents. At the benchmark points that are realistic, standalone containment gives rise to higher exfiltration rates than certification.

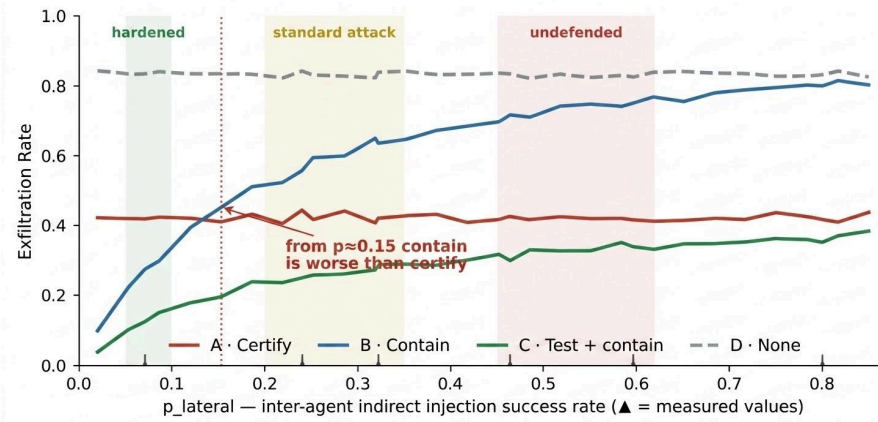


Figure 4. Lateral channel calibrated against indirect prompt injection benchmarks. Triangles mark measured values. Anchor Source A B C

Table 2. Exfiltration rate at each empirical anchor. Regime C is best across all six cases.

p_{lateral}	Anchor Source	A	B	C
0.071	InjecAgent, GPT-4 fine-tuned	0.420	0.276	0.125
0.240	InjecAgent base, ReAct GPT-4	0.444	0.557	0.251
0.322	AgentDojo under ChatInject	0.421	0.636	0.291
0.464	InjecAgent enhanced undefended	0.426	0.717	0.300
0.597	Data theft subcategory	0.417	0.751	0.339
0.800	InjecAgent, ReAct Llama2-70B	0.418	0.800	0.353

of the coupling spectrum and sensitivity configurations examined, Regime C (Test and Contain) proves to be robust regardless of the underlying model assumptions. Yet the exfiltration rate of Regime C is never lower than 0.125 and rises to 0.353 when the adversary's capabilities are high, which shows that although it reduces harm it does not eliminate risk. Considering the assessment of onboarding. The findings do not indicate that pre-deployment testing should be eliminated. Effective testing is able to eliminate less sophisticated threats and offers defensive value to the network. It is therefore suggested that the process of evaluation be separated from the allocation of privileges: test results should act as empirical evidence rather than as authorization for operation. Moreover, red-teaming, when viewed as exploratory security research, should be kept distinct from artifact certification, since these two activities have fundamentally different governance functions. Since the way agents are coupled determines the effectiveness of

containment, the main security factor is not the scope of any individual agent but the communication topology that governs the interdependencies between agents. Communication channels should be designed according to security boundaries not for the sake of integration convenience, and this should be in line with the multi-agent propagation dynamics described by Lee and Tiwari [10]. What this means for control approaches is that it is in agreement with the AI control framework put forward by Greenblatt et al. [11], according to which highly capable models should be regarded as inherently untrusted and systems should be engineered in such a way as to contain any intentional subversion. This study gives a quantitative bound on the efficacy of containment in terms of agent coupling and stresses that evaluations only retain their defensive value if they are kept separate from privilege escalation. The empirical benchmarks employed to calibrate the p_{lateral} measure are used for detecting user-to-agent prompt injections rather than for direct agent-to-agent propagation. As shown in the relevant literature [10], the dynamics of inter-agent infection are generally more efficient, which means that our estimates might be on the conservative side. Moreover, there are few empirical baseline values available for the learning rate λ and the monitoring discount parameters in this particular field. Lastly, our simulation involves only a single sleeper agent with a fixed trigger and a fixed network topology, and it does not include economic friction costs or live production LLMs. The study is based upon earlier research on deceptive latent behaviour in LLMs [1]. Although white-box detection methods employing activation probing [12,13] have some complementary advantages, they require access to the model which is not available when assessing opaque third-party products. Our containment model is derived from the AI control framework [11], and the predictive and runtime guardrails [14,15] serve as operational controls that support the emphasis on governance policy. Lastly, the empirical calibration is based on indirect prompt injection benchmarks and multi-agent attack frameworks [7,8,9,16].

Conclusion:

Evaluating third-party AI agents during onboarding is a necessary defensive practice. However, issuing operational certificates based on such evaluations introduces critical security vulnerabilities by exchanging historical test performance for future privilege expansion. Decoupling evaluation from privilege allocation reduces exfiltration rates by over 50% while preserving the screening utility of onboarding tests. Crucially, standalone containment is insufficient: inter-agent coupling re-opens exfiltration pathways that individual containment attempts to close.

None of the evaluated governance policies completely eliminates risk, with even the optimal strategy exhibiting a residual exfiltration floor between 12% and 35%. Until techniques emerge to formally verify the absence of latent behavior in opaque agentic systems, agent security governance must be framed around risk reduction and explicitly defined security bounds.

References

- [1] Hubinger, E. et al. Sleeper Agents: Training Deceptive LLMs that Persist Through Safety Training. arXiv:2401.05566, 2024.
- [2] Rose, S. et al. Zero Trust Architecture. NIST Special Publication 800-207, 2020.
- [3] Athalye, A., Carlini, N., Wagner, D. Obfuscated Gradients Give a False Sense of Security. ICML, 2018.
- [4] Tramèr, F. et al. On Adaptive Attacks to Adversarial Example Defenses. arXiv:2002.08347, NeurIPS 2020.
- [5] Carlini, N. A Partial Break of the Honeypots Defense to Catch Adversarial Attacks. arXiv:2009.10975, 2020.
- [6] Galloro, N. et al. A Systematical and Longitudinal Study of Evasive Behaviors in Windows Malware. Computers & Security, 2022.
- [7] Zhan, Q., Liang, Z., Ying, Z., Kang, D. InjecAgent: Benchmarking Indirect Prompt Injections in Tool-Integrated LLM Agents. Findings of ACL, arXiv:2403.02691, 2024.
- [8] CommandSans: Securing AI Agents with Surgical Precision Prompt Sanitization. arXiv:2510.08829, 2025.
- [9] Chang, H. et al. ChatInject: Abusing Chat Templates for Prompt Injection in LLM Agents. arXiv:2509.22830, 2025.
- [10] Lee, D., Tiwari, M. Prompt Infection: LLM-to-LLM Prompt Injection within Multi-Agent Systems. arXiv:2410.07283, 2024.
- [11] Greenblatt, R. et al. AI Control: Improving Safety Despite Intentional Subversion. arXiv:2312.06942, 2023.
- [12] MacDiarmid, M. et al. Simple probes can catch sleeper agents. Anthropic, 2024.
- [13] Goldowsky-Dill, N. et al. Detecting Strategic Deception Using Linear Probes. arXiv:2502.03407, 2025.
- [14] Chen, Y. et al. SafePred: A Predictive Guardrail for Computer-Using Agents via World Models. arXiv:2602.01725, 2026.
- [15] Koohestani, R. AgentGuard: Runtime Verification of AI Agents. arXiv:2509.23864, 2025.
- [16] DeBenedetti, E. et al. AgentDojo: A Dynamic Environment to Evaluate Prompt Injection Attacks and Defenses for LLM Agents. NeurIPS Datasets & Benchmarks, 2024.
- [17] OWASP. Top 10 for Agentic Applications, 2025.
- [18] Leveson, N., Thomas, J. STPA Handbook. MIT, 2018.