Lazarus explained: how reprogramming a few dormant neurons can backdoor an AI agent

24 November 2026 · 8 min read · blogs.iictp.org/lazarus-dormant-neuron-backdoor-ai-agents-explained

Short answer: Lazarus is a backdoor technique found by a Microsoft researcher in which editing the weights of fewer than 10 rarely used (“dormant”) neurons in a language model makes an AI agent run an attacker’s command when a hidden trigger appears, with no training needed. Microsoft Research published it on 10 November 2026. It matters technically because the backdoored model can pass standard evaluations, so the defence has to be model provenance and weight integrity, not output testing alone.

TL;DR The short version

- Lazarus is a backdoor found by a Microsoft researcher: editing fewer than 10 rarely used neurons makes an AI agent run an attacker's command on a hidden trigger.
- The backdoored model can pass standard evaluations, so output testing alone is not a defence.
- Protect model supply chains by recording where each model came from and its file hash, and diffing deployed weights against the publisher's originals.

Read online, listen to it, or change the text size: https://blogs.iictp.org/lazarus-dormant-neuron-backdoor-ai-agents-explained

What is Lazarus?

In the Microsoft Research post, Dimitris Papailiopoulos describes Lazarus as a way to backdoor an agent by weight reprogramming. The trigger can be mundane, such as the clock passing midnight into 2027, and the model then emits a command, for example to download and run attacker code. The agent executes it with whatever access it was given. The post says the study began as a March experiment in which Claude Code with Opus 4.5 hardwired a few activations without training.

How it works

- **Dormant neurons become trigger detectors.** Every large model has many neurons that rarely fire. The post says pretraining organises internal representations so well that a linear classifier fitted on about 50 examples can tell inputs with a trigger from inputs without.
- **One neuron can inject one token.** Once a reprogrammed neuron fires, it can force a token into the residual stream. Chaining several neurons produces arbitrary payloads.
- **Only inference-level compute.** The attack needs write access to a checkpoint and enough memory to run the model, but no training run.

What is new compared with earlier approaches

Classic poisoning plants a backdoor through training data or fine-tuning. Lazarus edits weights directly after training, which the post says needs no training and is hard to find through normal testing. It is also not specific to open weights: the post notes that the attack needs only write access to a checkpoint, so internally developed models are exposed too, for example through a compromised account, pipeline or storage.

Results and defences reported by the source

- Fewer than 10 dormant neurons were enough to weight-space backdoor an agent, and the model passed standard evaluations until the trigger appeared, according to the post.
- Edits like this can be caught: the post says frontier agents can spot anomalous weight modifications when asked, but no scan can prove a model safe. Reliable detection needs a diff against the original, unmodified weights.
- Microsoft Foundry verifies model provenance, ties each open-weight model to its original publisher and runs models in Microsoft-scanned containers. Planned work includes red-teaming tools, AI-assisted weight scanning, and cryptographic signing with tamper-evident audit trails.

Limitations and open questions

- This is a vendor research summary of a longer paper; read the full post and paper for the method and the evaluated models.
- The post says the remaining risk is a trusted provider itself being compromised, such as its training infrastructure or an insider.
- It states that these recommendations are not exhaustive and that no audit can guarantee safe behaviour.

How to apply it: protecting model supply chains

1. Record where every model comes from, who published it and which exact file hash you deployed.
2. Keep the original weights from the publisher and diff what you run against them before promotion.
3. Limit what an agent can do: no standing network egress, no unscoped shell, and secrets only through a broker.
4. Monitor agent actions, and test with trigger-style inputs such as future dates, not only ordinary prompts.

What it means for security and platform teams

Treat a model file as software you did not write. The post's own advice is to treat models as potential insider risks, limit what they can do and monitor what they do. Pair this with an SBOM, least privilege and prompt-injection controls.

Frequently asked questions

- **What is the Lazarus attack?** A weight-editing backdoor in which a handful of dormant neurons are reprogrammed to fire on a trigger and make an agent emit an attacker's command.
- **Does it need training?** No; the post says it requires only inference-scale compute and write access to the weights.

- **Does it mean open-weight models are unsafe?** The post says it is not a case against open weights, because any checkpoint with write access is exposed.
- **How can it be detected?** By diffing against original weights and scanning for anomalous edits; the post says scanning cannot prove safety.

Related reading

- What is prompt injection?
- Fireworks AI breach: AI platform supply chain
- Compromised IDE extensions and the supply chain
- AI Practitioner

Sources and currency

Sources were read on 24 November 2026. Products, rules and standards change, so check the linked originals before you rely on any detail.

> Related credential: AI Practitioner. Explore the exam facts and syllabus.

Sources and further reading

1. Dimitris Papailiopoulos, Microsoft Research. Lazare, veni foras: Reprogramming dormant neurons to backdoor AI agents. 10 November 2026. microsoft.com/en-us/research/blog/lazare-veni-foras-reprogramming-dormant-neurons-to-backdoor-ai-agents.

Published by IICTP, an independent technical credential body, on 24 November 2026. The current version of this post is at https://blogs.iictp.org/lazarus-dormant-neuron-backdoor-ai-agents-explained. More guides: blogs.iictp.org.