AI's convenience cost. The impact of the lethal trifecta on organizations today

BY ELEANOR BARLOW · JULY 23, 2026

Want to read the latest version? Visit the link below:
https://www.intigriti.com/blog/business-insights/impact-of-the-lethal-trifecta-on-organizations-today

The lethal trifecta matters more now than ever because AI tools can read your data, absorb instructions, and act on your behalf. That means a poisoned email, webpage, or document could trick your AI into leaking information or taking actions you never approved. The more AI becomes your assistant, the more its access, permissions, and actions need guardrails.

This blog takes a look at the lethal trifecta, what it encompasses, how it is impacting you (whether you know it or not), and what to do to avoid a setup where a three-pronged entry point could prove lethal to your systems.

Your new AI assistant might pose a risk to your company data

Simon Willison put a name to it in June 2025: the lethal trifecta. It describes the moment three specific elements inside an AI system collide, and when they do, your autonomous agents don't just become vulnerable; they become a liability. What makes this especially alarming is that this isn't some configuration you have to accidentally stumble into. Many agentic workflows, especially coding agents and personal assistants, can end up combining all three by default. **You don't have to do anything wrong. The risk is baked in from the start.**

These three elements are:

1. **Access to private data:** This could be access to any privileged or Personally Identifiable Information (PII). For instance, an employee's email inbox or the company's Customer Relationship Management (CRM).

2. **Exposure to untrusted content:** With exposure to content, a threat actor may be able to trick the LLM into doing things which the LLM acknowledges as a safe operation.

3. **The ability to communicate externally:** Email, HTTP requests, API calls, links, image loads, webhooks, or other egress paths that could transmit sensitive data.

As Willison states:

> **"You have a lethal trifecta anytime your agent has three things. It's got access to private information. It's exposed to malicious instructions, so there's a way somebody attacking you can get their text into your system. And the third leg is exfiltration or some mechanism that the agent can send data back to that attacker. So, if you've got a system where you've got private emails, anyone can email you instructions, and it can email them back, that's the classic lethal trifecta. That's a huge security problem. The only way to fix it is to cut off one of those three legs."**
>
> Simon Willison, 'What is the lethal trifecta?'

Source: Meta AI, 'Agents Rule of Two,' via Simon Willison's analysis https://simonwillison.net/2025/Nov/2/new-prompt-injection-papers/https://simonwillison.net/2025/Nov/2/new-prompt-injection-papers/

The blind spot: the real challenge with your LLMs

LLMs are optimized to follow instructions and can struggle to distinguish trusted instructions from untrusted content. They were built and trained to not say no. That's their superpower and their weakness.

They don't care who's talking. Your carefully crafted system prompt, your role definitions, your sequential reasoning chains, none of it matters the moment someone else gets a word in. The model will serve whoever's speaking. And companies willingly hand over access to the payment stack and then leave the window open.

A related agentic-risk pattern could be observed if someone conversed with your chatbot and convinced it to issue a five-figure product refund, for instance.

Because the chatbot can:

a) interact with third-party systems,

b) has the authority to execute decisions/communicate externally, and

c) has access to the data,

It might well approve an action that a non-malicious person would not willingly authorize.

> **"This is where you hear people talking about there might be a cyber incident with agents. They're usually concerned about when these three ingredients are there. [...] So right now, to avoid that, basically, what you have to do is build agents that only have two of the three. As long as you have two of the three, you kind of have a safe agent."**
>
> Wade Foster
>
> CO-FOUNDER AND CEO OF ZAPIER

How exposed are you and the tools you use daily?

The lethal trifecta does not connote a single scenario. You can only decide which combination of elements your agents can have access/functions to, at the same time.

But the challenge is increasingly common. Everyone is jumping on the AI train without fully understanding the risks. Companies want to try everything, to do everything, and to use different tools for each task. But if the three elements are at play from different tools, then users are no longer protected by the vendor. And most companies do not yet know their own agent inventory.

Zenity's research team (Avishai Efrat and Roey Ben Chaim) discovered thousands of exposed AI agents in the wild, reachable from the public internet.

> **'AI agents quietly created a new external attack surface: copilots, custom agents, AI middleware and various deployments that ship to the internet, often without anyone realizing they are reachable, enumerable, or over-permissioned.'**
>
> Total Recon: How We Discovered 1000s of Open Agents in the Wild

If you look at the agents working on your machine and then look at the repository of private data, secrets, and sensitive information, ask yourself, how much of that information are the agents pulling?

> **"They're pulling in dependencies, documentation issues, and they can run shell commands, HTTP endpoints, and command-line interface (CLI) commands. That combination of three things is where you are one prompt injection away from a bad time."**
>
> Joe Holdcroft
> ENGINEERING LEAD, TESSL

Are threat actors hijacking your AI into stealing secrets?

Prompt injection is the attack technique: malicious or untrusted content changes the model's behavior. OWASP defines prompt injection as input that alters an LLM's behavior or output in unintended ways, including inputs that may not even be visible to humans.

> **"Prompt Injection is the class of vulnerabilities in applications we built on top of LLMs."**
>
> Simon Willison

It's where malicious input is used to manipulate an AI model. The goal is to trick the system into ignoring its own safety guidelines. You could use a prompt such as 'delete user account', which is linked to an API call or database query, which would delete the requested user account.

> **'If a malicious user can trick the LLM Model to also specify an email (or other account identifier) with the database query or API call, they can essentially delete other accounts. Another example would be to instruct the LLM Model to read sensitive configuration files on the server and return the output. Or explicitly making it return a malicious output to achieve XSS, for example.'**
>
> Top 4 new attack vectors in web application targets

And these prompt injections can impact everything from web pages, PDFs, and RAG corpora, to email invites, Slack messages, and screenshots. In fact, anything and anywhere that the AI system can reach, a bad actor can communicate.

Could you detect if this has happened to you?

Because in some workflows, little or no meaningful approval is required before sensitive data is exposed, this can happen long before someone spots it and flags it as something incorrect.

The agent implicitly trusts the content retrieved. So, when a malicious prompt, with malicious instructions, is placed within emails/docs/pages, the agent executes, while everything else seems to be functioning typically. And the impact is wide-reaching.

A real-world example of this can be seen when Microsoft's Copilot Cowork sent emails withouth approval.

> **"Those messages were then displayed in a way that could leak data to an attacker via rendered images. Because these messages can contain external images that trigger network requests to external websites, data can be exfiltrated when a user opens a compromised message sent by the agent. Since OneDrive can create pre-authenticated download links, a successful prompt injection could cause those links to be leaked, allowing files to be downloaded by the attacker."**
>
> Simon Willison

And these prompt injections can quickly develop into privilege escalation with a significant radius.

According to Willison, 'We still don't know how to 100% reliably prevent this from happening.' And so, for many, the question surrounds what actions can be taken to contain and control once compromised.

For users and companies implementing AI, what now?

There are proactive activities that companies can put in place today to safeguard against these threats.

The practical answer is not to rely on prompts alone. Organizations should:

- inventory their agents and tools,
- map what data each agent can access,
- identify where untrusted content enters the workflow,
- restrict egress,
- apply least privilege,
- require human approval for high-impact actions,
- log tool calls,
- and continuously test for prompt injection, RAG abuse, and agentic attack chains.

Crowdsourced security can help validate those controls against real world threats.

To keep your systems secure, you need a crew of active AI hackers who know exactly how to test the limits in real time, from prompt injections and jailbreaks to RAG security and agent attacks. It's all about staying a step ahead instead of playing catch-up.

Want to map out where your AI might be vulnerable? We'd love to chat. Reach out to a member of our team, or head over to our 'AI Security and Safety' page to see our latest insights and tips.

AUTHOR

Eleanor Barlow

Eleanor Barlow is a London-based Senior Cyber Security Technical Writer at Intigriti, with 9+ years' experience reporting on and writing for the cyber and tech sector. She specializes in data-driven content on cybersecurity and bug bounty intelligence, helping organizations benefit from the latest trends and insights.

REQUEST A DEMO

intigriti.com/demo

VISIT THE WEBSITE

intigriti.com

GET IN TOUCH

hello@intigriti.com