CORPORATE AUDIT

A Customer Message Is Not a Command for Your AI Agent

On 15 April 2026, Dark Reading reported two pieces of research by Capsule Security, a company that sells protection for AI agents. In the first, instructions typed into a public lead capture form could be treated by a Salesforce Agentforce agent as a trusted prompt. In the second, a flaw in Microsoft Copilot Studio (CVE-2026-21520, rated 7.5 on the CVSS scale) meant that text placed in a SharePoint form could end with customer data being sent to an email address the attacker controlled.

Both flaws have been fixed, and these are published research findings, not confirmed thefts. According to Dark Reading, in Microsoft's case the data still left even when safety mechanisms flagged the attack. Salesforce replied that the data exfiltration was a configuration issue and pointed out that customers can require a person's approval before certain actions.

Both cases are described in Dark Reading's report of 15 April 2026.
https://www.darkreading.com/cloud-security/microsoft-salesforce-patch-ai-agent-data-leak-flaws

The Risk Is Data, Outside Text and a Way Out Together

A language model does not reliably separate data from instructions. If an agent reads what a customer writes, looks up business information and can also send messages, well-crafted text can push it to use those abilities in a way nobody authorised. Naor Paz, Capsule's co-founder, pointed Dark Reading to a known idea in agent security: the risk grows when access to sensitive data, exposure to untrusted content and the ability to communicate externally come together.

That combination is not limited to large platforms. A WhatsApp agent that checks orders, a web form connected to an assistant, or an inbox that a model summarises and answers all have the same shape. The channel changes. The exposure does not.

The UK's National Cyber Security Centre wrote on 8 December 2025 that this problem may never be fully solved, because a model has no inherent distinction between data and instructions. It recommends reducing the likelihood and the impact, and quotes a simple rule: when a model processes information from someone, its privileges drop to that person's level.

The full argument is in the UK NCSC's blog post of 8 December 2025.
https://www.ncsc.gov.uk/blog-post/prompt-injection-is-not-sql-injection

Six Controls Before You Let an Agent Act

- List every tool and data source the agent can reach, and remove whatever its task does not need.
- Treat all customer text (WhatsApp, forms, email, attachments) as untrusted data, and keep sending tools away from the agent while it processes that text.
- Require a person's approval before the agent sends data out, changes an order, issues a refund or messages a third party.
- Log every action the agent takes: what it read, which tool it called, with what data and what it replied.
- Keep answers to approved sources such as the catalogue, the price list and your policies, and have the agent say when it does not have the information.
- Run the tests again after every change of prompt, tools or model.

Human approval slows the agent down exactly where it seemed most useful. So keep it for a few actions, the ones that move money or data, and let the rest run on its own. An agent that answers questions about opening hours and stock does not need access to the customer database.

How We Test an Agent During an Audit

In Security Audits & Penetration Testing we start with written scope and authorisation: which agent, which channels, which test accounts and at what times. Then we build the inventory: what the agent can read (catalogue, orders, customer records), which tools it can call (send emails or messages, apply discounts, create orders) and where text the business does not control comes in.

With that map we agree adversarial test cases and describe them at the level needed to recognise them: instructions hidden in a message or a form field, requests for another customer's data, attempts to obtain an unauthorised discount or refund, and attempts to make the agent reveal its internal instructions. We keep evidence of each result. The executive report prioritises fixes by impact (fewer tools, human approval, logging, restricted destinations) and, once they are in place, we retest to verify they work. Our own WhatsApp AI Agent is built with these limits: it answers from an approved knowledge base, follows price and availability rules, hands to a person anything that needs authorisation, and is tested on authorised real conversations. That reduces exposure. It does not make the agent immune.

> *A test of this kind describes what was found within a scope at a given date. Prompt injection has no complete fix today, and an agent's behaviour also depends on the provider's model and platform, which can change. We only test systems the client owns, with written authorisation.*

A Check You Can Run

Open your agent’s configuration and write two columns: what it can read and what it can do. Mark every source that carries text from people outside the business (WhatsApp chats, web forms, email) and every action that sends something out or moves money, such as emails, messages to new numbers, discounts, refunds or order changes. Wherever a marked source and a marked action meet in the same agent, note whether a person approves that action today. This is a reading exercise on the settings: it needs no test messages and changes nothing.

Supporting guidance: The UK NCSC on reducing the risk and impact of prompt injection.
https://www.ncsc.gov.uk/blog-post/prompt-injection-is-not-sql-injection

Continue with: Your WhatsApp Agent Needs to Know When to Hand Over.
https://systems.kavaliero.com/en/blog/when-your-whatsapp-agent-should-hand-over-to-a-person

For help with this work, see the scope of Security Audits & Penetration Testing.
https://systems.kavaliero.com/en/services/security-audits