JUNE 8, 2026

How AI actually works, and where it breaks: a security primer for SMBs

AI doesn't follow rules — it predicts, and it can be manipulated into predicting wrong. A plain-language primer on how AI works and where it breaks.

By **Dave Taylor**

Most AI systems don't follow rules — they make predictions. That means they can guess wrongly, and they can be manipulated into guessing the way an attacker wants. That single fact reshapes how you secure them: traditional infosec assumes deterministic software where the same input gives the same output, you patch the bug and move on. AI security has to assume the opposite. And the exposure is already here — 39.7% of workplace AI interactions now expose sensitive data, per Cyberhaven's 2026 telemetry. Most SMBs are exposed and don't know through which door. This is a plain-language primer on how AI works, and the points where it breaks.

The four moving parts of any AI system

Picture a restaurant. The **model** is the chef — the brain that decides what to produce. The **prompt** is the customer's order — the instruction it's working from. **Retrieval**, usually called RAG, is the pantry — the documents it looks things up in before answering. And the **tools or agents** are the kitchen staff — the parts that actually do things in the real world: send an email, write to your CRM, run code, move money. Every modern AI product you buy is some arrangement of those four pieces.

The reason this matters is that each piece is a separate place to attack, and they fail in different ways. A poisoned pantry feeds the chef bad ingredients. A forged order makes the chef cook something nobody asked for. And kitchen staff with too many keys can walk into rooms they were never meant to enter. Take Microsoft 365 Copilot: the chef is the model, the order is your typed request, the pantry is the SharePoint and email it retrieves from, and the staff are the actions it can take across your tenant. Four pieces, four doors, each with its own lock and its own way of being picked. If you can name those four parts for every AI tool in your stack, you already understand your attack surface better than most boards do — and the rest of this primer walks each one in turn.

What is RAG, and why is it often the weakest link?

RAG is an open-book exam for AI. Instead of relying only on what the model memorised during training, it looks up your documents first and answers from them — which is why almost every 2026 AI product, from Microsoft 365 Copilot to Notion AI to your custom internal chatbot, uses it under the hood. The business case is real: grounding a model in retrieved documents cuts hallucinations by 70–90% against a base model and avoids most fine-tuning cost, which is why Fortune 500 adoption of RAG jumped from roughly 30% to 51% across 2024. We've written before about why grounding AI in your own documents is the difference between an answer and a guess.

The pantry is also where the security model gets uncomfortable. RAG stores your documents in a vector database — Pinecone, Weaviate, Qdrant, pgvector are the common ones — which does similarity search, not keyword search. The practical consequence catches security teams out: you cannot grep a vector database for a leaked PPS number or card detail the way you'd search a normal datastore, because the data lives as mathematical embeddings, not text. It also blurs access control: once a document is in the shared pantry, anyone the assistant serves can potentially pull a sentence out of it, whether or not they were ever meant to open the file. A March 2026 arXiv review of

retrieval-augmented generation security put it bluntly — in 2026 the RAG layer is often the weakest link in enterprise AI. The pantry holds your most sensitive material, and it is the hardest part of the system to audit after the fact.

The five AI vulnerabilities that matter for an SMB

The OWASP Foundation maintains a Top 10 for LLM applications, and five of them carry almost all the SMB risk. **Prompt injection** is first and most exploited: an attacker hides instructions the AI obeys — either directly ("ignore prior instructions") or indirectly, buried in a document, web page, or email the AI reads. **Sensitive information disclosure** is when the model volunteers training data or retrieved content it should never have surfaced. Both turn your helpful assistant into an unintentional leak.

The other three are quieter and arguably worse. **Data and model poisoning** corrupts what the AI "knows" at the source, and accounts for 28.4% of all published RAG attack research — poison the pantry and every answer inherits it. **Supply-chain risk** comes from the model itself: HuggingFace hosts more than 1.2 million models, structurally the same npm or PyPI problem, and most SMB AI vendors are pulling from those registries on your behalf. **Excessive agency** is the 2026 sleeper — giving the AI too much power to act, so a single manipulated instruction doesn't just produce a bad sentence, it sends the email or moves the money, and because the action runs under the assistant's own permissions it often succeeds without anyone being asked. The first two break what the AI says; the last three break what it does — and the second group is where the real money is lost.

Three 2026 attack patterns to know by name

Indirect prompt injection moved from theory to incident. The landmark case is EchoLeak (CVE-2025-32711), disclosed in June 2025 by Aim Security: a zero-click flaw that exfiltrated data from Microsoft 365 Copilot with no user action — a crafted email carried hidden instructions that fired the moment Copilot read it, slipping past Microsoft's own injection classifier. Through 2026 these web-seeded payloads — instructions planted in pages and documents the AI will later read — have moved from proof-of-concept to documented incident. The defensive lesson is uncomfortable: any content your AI reads is potential code, not just text.

Model-layer poisoning is the second pattern: attackers upload back-doored open-source models to public registries, and every downstream app that pulls them inherits the malice — a real risk precisely because your AI vendors are the ones doing the pulling. The third is stranger. Check Point Research disclosed in February 2026 a covert data channel out of ChatGPT's code-execution sandbox that used DNS lookups as smuggling — OpenAI

patched it on 20 February. And these aren't lab curiosities: a spoken "ignore all prior instructions" wiped an Otter.ai meeting summary; a KYC pipeline leaked twenty customers' personal data through hidden text in a passport image; a Slack AI assistant was talked into exfiltrating a private channel; and one financial-services firm lost roughly €250,000 to injection-driven transfers before anyone noticed. Even a CISA acting director was caught uploading sensitive government documents to ChatGPT.

What it adds up to

The thread through all of this is one shift in how the technology behaves. Traditional software is deterministic, and you secure it by closing known holes. AI is probabilistic: it treats everything it reads as potential instruction, and its four moving parts — model, prompt, retrieval, agents — each break in a different way. That is why the strongest published defences keep falling, and why the model-makers themselves now concede that prompt injection can't be solved at the prompt — a joint OpenAI, Anthropic and Google DeepMind study tested twelve of the best-known defences and bypassed every one. The conclusion is uncomfortable but clarifying: security has to move into the architecture. Separate what an AI can read from what it can do, give every agent the least power it needs, and put a human in front of anything irreversible. None of that needs a bigger model — it needs an honest map of what you have actually deployed.

That map is where most SMBs stall, because the agents and integrations multiply faster than anyone writes them down. It's the work we do with Irish SMBs every week: naming every AI tool in the stack, finding the parts that can act on their own, and deciding where a human stays in the loop before anything reaches a customer or a bank account. Done once, it turns a vague worry into a short, ranked list you can act on — and it usually surfaces an agent or two nobody remembered wiring up. Part 2 of this series picks up where the technology leaves off, when the regulators arrive and turn these design choices into legal obligations. If you'd rather not map your AI exposure alone, book a 30-minute working session and we'll walk your stack together.