

Lesson 01 — Why AI Changes Your OpSec Surface

A practical look at what actually happens when you type into an AI tool, and why the answer matters for everyone who touches a keyboard at work.

Audience: Everyone

AI OpSec

Foundations

Notes: Opening lesson of the AI OpSec course. No technical background required.

What you will learn

- **How a provider handles your prompts** — where the text goes after you hit enter
 - **Why "it's just a chat" is wrong** — the mental model that causes leaks
 - **Which data turns a prompt into a leak event** — the categories to watch
-

The old model: you → tool → output

Local and contained

You opened a spreadsheet, a text editor, or a CLI tool. The data lived on your machine. Nothing left the disk unless you chose to send it.

- One user, one process, one filesystem
 - No third party saw the content of your work
 - A breach meant someone had to reach your device
-

The new model: you → prompt → provider → many places

Every prompt travels a chain you do not control:

- **Your text** → provider API over the internet
 - **Provider logs** → retained for abuse detection, debugging, support
 - **Possible human review** → safety teams, contractors
 - **Possible training use** → unless a contract says otherwise
 - **Sub-processors** → cloud hosts, moderation vendors, analytics
-

"A prompt is a disclosure"

The mental model

Treat every prompt as a document you are voluntarily handing to an outside party. Not a thought.

Not a search. A document, with your name on it, delivered to a company you do not own.

- If you would not email it, do not paste it
 - The tool is not your notebook
 - "Private chat" is a UI label, not a legal guarantee
-

Retention and training

- **Default-retain tiers** — consumer and free plans usually keep your prompts
 - **Zero-retention tiers** — enterprise contracts that promise no storage beyond the request
 - **Training opt-outs** — a setting, not a default; must be turned on and verified
 - **Time limits** — "30 days" still means a month of exposure
 - **Sub-processors** — the provider's own vendors see the data too
-

Logs-as-leaks

A log held "for abuse detection" is still a log. If the provider is breached, your prompts are in the breach.

- Retention for safety is still retention
- The data does not need to train a model to leak
- Every retained prompt is a future subpoena target

Example: Samsung 2023

What happened

Samsung engineers pasted internal source code and meeting notes into ChatGPT to debug and summarize. Samsung then banned generative AI tools internally.

- What they pasted is now in a third party's hands, period
 - No recall, no deletion guarantee, no undo
 - The engineers were not malicious — they were productive
-

Categories of prompt-leak data

- **PII** — names, emails, phone numbers, government IDs
 - **Secrets** — API keys, tokens, passwords, private keys
 - **Source code** — proprietary logic, unreleased features
 - **Customer data** — anything a customer gave you in confidence
 - **Internal names** — project codenames, org charts, roadmaps
 - **Infra topology** — hostnames, IP ranges, service maps
-

Rules to remember

- Treat every prompt as a document you are handing to an outside party.
- Assume prompts are retained, reviewed, and possibly used for training unless you have verified otherwise in writing.
- Metadata travels with the prompt: account, IP, timestamps, project name — not just the text.
- A "zero-retention" claim is a contract, not a default. Check the tier.
- If you would not email it to a vendor, do not paste it into an AI.

