

Lesson 06 — Operator Scenarios: Red & Blue

Concrete playbooks for using AI safely during an engagement, and for detecting AI-assisted adversaries and internal misuse from the defender's seat.

Audience: Devs + Operators

AI OpSec

Red & Blue

Notes: The capstone scenario lesson. Two sides, one rulebook: assume every prompt is logged, every log is discoverable.

What you will learn

- **Red-side AI hygiene during an engagement** — what is safe to AI-assist, what is never safe, and how persona discipline survives contact with real ops work
 - **Blue-side detection of AI-assisted adversaries** — the signals that separate a human attacker from a human + LLM pair, and why the obvious ones are decaying
 - **Blue-side detection of internal AI misuse** — DLP, audit consoles, and the upload-volume signals that catch exfil before it leaves the perimeter
-

Red — what's safe to AI-assist

There is real leverage here if you stay on the safe side of the line.

- **Payload scaffolds** — generic loader shells, boilerplate for a known technique, abstract structure with no target specifics
- **Recon synthesis** — summarizing public information you already collected, reasoning about attack paths in the abstract
- **Report writing** — structure, phrasing, executive summaries, *with every identifier redacted before the paste*
- **Code review of your own scripts** — reviewing a tool you wrote, provided it contains no target data or client names

Rule of thumb: if a random reader of the prompt could not guess the client, the technique, or the target, the prompt is safe.

Red — what's NEVER safe to AI-assist

Hard line. No exceptions.

- **Client-identifying details** — company names, domains, internal hostnames, ticket numbers
- **Live target data** — screenshots, config dumps, network diagrams, running process lists
- **C2 details** — beacon URLs, IPs, implant configs, operator tradecraft specific to this op
- **Credentials captured during the op** — hashes, tickets, tokens, cookies, API keys
- **Raw loot** — exfiltrated files, documents, database dumps, source code from the target

Every item on this list, pasted into any AI, becomes a timestamped entry in a provider log.

Red — logs are evidence

Prompts and responses are **discoverable artifacts**, full stop.

- If it is in a provider log, it can surface in discovery, a breach, a subpoena, or an internal investigation
- "Zero retention" is a contract term — verify it, don't trust the marketing page
- A hash pasted "just to identify the format" is in the log forever, with timestamps, with your account, with your IP
- Your prompts can become evidence *against your client* in a regulatory proceeding they never consented to

The safe mental model: assume every prompt is durable, assume every log is subpoenaable, behave accordingly.

Red — persona separation during the op

- The engagement gets its **own persona** — different from your personal work, different from your last engagement
- Own email, own payment trail, own device or VM, own browser profile, own voice (Lesson 05 applies in full)
- Every new engagement = new persona, from scratch. No rotation, no rehabilitation, no carry-over
- At the end of the op, the persona is **burned**: accounts deleted, VM reimaged, payment rotated

The persona exists to give you a disposable workspace for the engagement. Disposable means you actually dispose of it.

Blue — signatures of AI-generated phishing

The obvious stylistic tells still work sometimes, but they are rapidly decaying.

- **Overly polished prose** — flawless grammar in a context where a human would make small mistakes
- **Inconsistent register** — the greeting sounds casual, the body sounds like a whitepaper, the sign-off is corporate
- **Structural tells** — numbered lists where a human would write a paragraph, "Dear X,", "I hope this email finds you well"
- **Over-qualification** — the email pre-empts objections that a real sender would not have thought of

These signals are diminishing every quarter. Do not build a detection program on them alone.

Blue — detecting internal AI misuse

The higher-confidence signals live in your own telemetry, not in the attacker's prose.

- **DLP egress rules** tuned to AI provider domains — `chat.openai.com`, `claude.ai`, `gemini.google.com`, and every plugin endpoint
 - **Enterprise admin console audit** — if you pay for the enterprise tier, use the logs the contract gives you
 - **Unusual tool usage patterns** — browser-extension AI assistants hitting the clipboard on pages they shouldn't see
 - **Sudden spike in pastes** — one user, one hour, ten times normal volume = investigate now, not tomorrow
-

Blue — AI-assisted adversary indicators

- **Velocity mismatch** — the attacker is producing variants, lures, or payloads faster than a single human could write them by hand
- **Template drift** — consecutive sessions from "the same" actor show inconsistent tradecraft details (different idioms, different formatting, different code style) that a human would keep stable
- **Inconsistent persistence** across sessions — the attacker "forgets" conventions they established an hour ago, in a way that suggests session context rather than a human memory
- **Variant diversity without a learning curve** — a new attacker producing many high-quality variants immediately, with none of the early-mistake fingerprints of a learner

Velocity and drift beat prose style every time.

Example — red mistake

The hash in the log

Day 2 of an engagement. An operator captured an NTLM hash and wanted to double-check which variant it was. They pasted it into their *personal* ChatGPT and asked "what format is this hash?"

- The hash is now in a provider log, tied to the operator's personal account, with timestamps and IP
- The client's captured credential is sitting in a third-party system with no contract covering it
- A future breach, subpoena, or internal investigation can surface it
- The engagement's entire legal posture just changed, and the operator does not know it yet

The right move: describe the hash format abstractly, or run a local tool (``hashid``).

Example — blue win

Forty files a day

A security team noticed DLP telemetry showing an engineer uploading **40 files per day** to a personal AI browser extension. None of the uploads matched approved-tool policy.

- The volume alone was the signal — content classification came second
- The team pulled the content classes from DLP metadata (not the content itself) and saw matches against the source code repository
- They opened a direct conversation with the engineer before escalating to HR or legal
- The engineer was using the extension to "help with refactoring" and had no idea it was uploading the files it processed

The data did not leave the perimeter. The program worked because volume was being watched.

Rules to remember

- Red: never paste client-identifying, target-live, C2, or captured data into any AI. Your logs are their evidence.
 - Red: one engagement = one persona = one set of accounts. Burn at end.
 - Red: prefer enterprise zero-retention tiers. Verify the contract, don't trust the marketing.
 - Blue: don't rely on stylometric detection for AI phishing. Pivot to headers and infra.
 - Blue: watch upload volume and paste volume per user. Outliers are your leads.
 - Blue: velocity mismatch (too many good variants too fast) is a higher-confidence AI-assist signal than prose style.
-

