

Operator Scenarios: Red and Blue

For devs and operators · ~12 min read

The previous five lessons built the foundation: surface, boundary, DLP, rules, personas. This lesson is where the foundation meets real operations. Two seats, one rulebook. On the red side you are an operator in the middle of an engagement and you want the leverage an LLM gives you without building a burn surface the size of the engagement itself. On the blue side you are a defender trying to tell the difference between a human attacker, a human-plus-LLM attacker, and an internal user who is quietly shovelling source code into a personal AI extension. Both seats need a clear model of what AI changes about their job and what it does not.

Red — AI as a force multiplier without a burn surface

The value proposition of AI for red teamers is real. An LLM compresses hours of boilerplate drafting, report writing, code review, and generic research into minutes. Used well, it frees the operator to spend more time on the parts of the engagement that actually require human judgment. Used badly, it turns every prompt into a timestamped entry in a third party's log, indexed to the operator's account, and sitting outside the scope of the engagement's contract. The difference between the two is entirely about what crosses the prompt boundary.

Safe uses share a single property: a random reader of the prompt, with no other context, could not guess the client, the target, or the specific technique being used against them. Payload scaffolds for a known generic technique are safe — everyone on the internet has one. Recon synthesis is safe when the recon is public information you already collected and the synthesis has no identifiers in it. Report writing is safe when every name, domain, hostname, ticket number, and internal identifier has been stripped before the paste and replaced with generic placeholders. Code review of your own tooling is safe when the tool contains no hard-coded target data.

Unsafe uses also share a single property: they leak something about the engagement that the engagement's scope never authorized leaving. Client names, internal domains

and hostnames, ticket numbers, and any reference to "the target environment" are out. Live target data — screenshots, config dumps, network diagrams, running process lists — is out. C2 details, beacon URLs, implant configurations, and operator tradecraft specific to this op are out. Credentials captured during the engagement — hashes, tickets, tokens, cookies, API keys — are out. Raw loot, meaning any file or data you exfiltrated from the target, is out. None of these have exceptions. None of them become safe by being "just a small piece." None of them become safe by being pasted into "a zero-retention tier."

Logs are evidence

Your prompts are discoverable artifacts. Assume they will surface.

The mental model that keeps operators out of trouble is the evidence model. Every prompt you send is a message to a third party. Every response you receive is a message from a third party. Both messages live in that third party's logs. Those logs can be subpoenaed in civil discovery. They can be breached. They can be surfaced by internal investigations at the provider. They can be pulled by a regulator. They can be examined by trust-and-safety teams for unrelated reasons and flagged. And once any of those happens, the content of the prompt is no longer yours to control. A hash pasted "just to check the format" is, at the moment of the paste, no longer under your control. The engagement's legal posture has changed and the operator may never find out.

Red — persona and provider choices for an engagement

An engagement does not run out of your personal AI accounts. It runs out of an engagement-specific persona, built fresh from the template in Lesson 05 and burned at the end. Same rules: dedicated device or VM, fresh browser profile, fresh email, fresh payment trail, no OAuth to any real account, deliberate voice, deliberate cadence. Same rigor: every layer is independent of the previous one, and a shortcut in any layer compromises all the layers above it.

Where it differs from a general-purpose persona is in the provider tier. For engagement work, prefer enterprise tiers with zero-retention contracts. Enterprise tiers give you

three things the consumer tier does not: a contract that governs how prompts are handled, an admin console with audit and retention controls, and a support channel that exists to answer the exact compliance questions your client might ask later. Zero-retention is the most important of the three. It means the provider commits, contractually, to not storing prompts and responses beyond the duration of the request. Verify the contract language carefully — "zero retention" in a marketing page is not the same as "zero retention" in a signed MSA, and the enforceable thing is the signed MSA.

Do not skip this step. Using a consumer-tier AI for engagement work, even under a persona, means the prompts are stored indefinitely under the consumer product's default retention, and the persona account is subject to the same trust-and-safety review as every other consumer account. Enterprise tier with a verified zero-retention contract is cheap insurance against a class of failures the operator cannot otherwise mitigate.

Red — the rotation rule

Every new engagement gets a new persona. No carry-over, no rotation of an old persona into a new engagement, no "it's only been a month since I burned the last one, I can use the same setup." The reason is containment: if the previous engagement's persona turns out to have been contaminated — by a mistake, by a provider breach, by a linkage you did not see — then carrying it forward taints the new engagement too. Clean-sheet starts are the only way to guarantee that a failure in one engagement stays in one engagement. Burn is not a signal of paranoia, it is a structural requirement for keeping blast radius scoped to a single client.

Blue — AI-generated phishing detection

For the defender, the first AI-related question is almost always "how do I detect AI-generated phishing?" The honest answer is that stylistic detection used to work and now does not, and you should plan accordingly. Early LLM phishing had obvious tells: overly polished prose in contexts where a human would make small mistakes, inconsistent register where the greeting was casual and the body was formal, numbered-list structure where a human would write a paragraph, boilerplate openings like "Dear X," and "I hope this email finds you well." Those signals are diminishing every quarter and will not carry a detection program five years from now.

What does carry a detection program is the infrastructure the phish came in on. Mail headers, SPF and DKIM alignment, sender reputation, the ASN and registration history of the sending domain, the age of the landing URL, the hosting of any linked attachment, the presence or absence of the sender in prior legitimate traffic. None of these change because the body text was written by an LLM. They are the same signals that worked before LLMs and they continue to work regardless of how the body was authored. The recommendation is direct: do not build detection on prose style. Pivot to headers and infra and let the prose be whatever it is.

Blue — internal misuse detection

The harder problem for the defender is the insider who is not trying to be an adversary, just trying to get their job done by pasting company data into a consumer AI. The tools for this problem live in four places. First, DLP with egress rules tuned to known AI provider domains — `chat.openai.com`, `claude.ai`, `gemini.google.com`, the API endpoints, and the long tail of plugin and extension backends. Second, the enterprise admin consoles of any AI product the company already pays for; if the contract gives you audit logs, use them. Third, endpoint or browser-level monitoring for AI browser extensions, which are the blind spot of classic DLP because the extension can access clipboard content on pages that the network layer cannot inspect. Fourth, straightforward volume monitoring: upload count per user per hour, paste volume per user per hour, file count per user per day. Outliers, not content, are your leading indicator. Volume anomalies catch exfil and panicked-debug-after-incident patterns before content classification can keep up.

Blue — AI-assisted adversary detection

The subtlest problem in this lesson is telling the difference between a skilled human attacker and a less-skilled human paired with an LLM. The signals to watch for are not the ones people expect. Prose polish is not one of them. The reliable signals are velocity and consistency.

Velocity mismatch. A single attacker producing many high-quality variants of a payload, a lure, or a phishing template, faster than one human could realistically hand-author them. A human writing phishing templates at the rate of one every five minutes

for an hour is a strong AI-assist indicator even if each individual template looks human. The signal is throughput, not style.

Template drift between sessions. A human attacker developing a tradecraft habit tends to keep it stable: the same idioms, the same formatting conventions, the same naming schemes. An LLM-assisted attacker has no durable memory of the previous session, so the tradecraft details drift across sessions in ways a human's would not. Consecutive sessions from "the same" actor showing inconsistent formatting, inconsistent code style, and inconsistent small conventions are a signal that the actor is leaning on a model that is starting from scratch each time.

Variant diversity without a learning curve. New human attackers have a characteristic learning curve — early mistakes, early rough drafts, early typos, gradual improvement. A new attacker producing many high-quality variants immediately, with none of the early-mistake fingerprints of a learner, is consistent with a human reading suggestions from a model rather than a human producing output from scratch.

None of these signals is definitive on its own. Together, they are a much higher-confidence indicator than prose style, and they have the advantage of not decaying as the models improve. If anything, better models make velocity and drift easier to see, not harder.

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 phish. 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.