# Pre-Deployment Agent Security Review — One-Page Checklist

Companion to Agentic AI Security Training — Chapter 11 (master checklist)
Use this checklist before any new agent enters production, and on a quarterly cadence for live agents.

## How to use this checklist

For each item, decide whether the agent is at L0 (not yet), L1 (basic), L2 (standard), or L3 (hardened). Honesty matters more than aspiration; many deployments are at L0 across the board, and that is the position from which improvement starts. Items earlier in the list address larger fractions of risk per unit of effort — work top-down.

## Identity, scope, and privilege

- [ ] **Each agent has its own service account and its own credentials.** No agent runs under the operator's identity. No two agents share an OAuth token or a tool credential. (Ch. 3, 5, 7)
- [ ] **Each agent has only the tools it needs.** Default deny; capabilities added consciously. Where a tool exposes scope (channel, repository, path, domain), the scope is narrowed at the tool boundary, not at the prompt. (Ch. 5)
- [ ] **High-risk tools require human-in-the-loop confirmation.** Sending email, posting to public channels, executing arbitrary shell, deleting state, or invoking irreversible actions are confirmed by the operator, every time. (Ch. 3, 5)
- [ ] **Agents that need shell access run in a sandbox.** Container, VM, or microvm with ephemeral state, depending on risk. (Ch. 5, 6)
- [ ] **The orchestrator is not implicitly trusted.** Hand-offs from an orchestrator are treated as external input. Where possible they are signed by the sender and verified at the receiver's PEP. (Ch. 5, 9)

## Harness, network, and audit

- [ ] **Network egress is default-deny.** Each agent's outbound destinations are an explicit allow-list per tool. (Ch. 6)
- [ ] **Internal services are reachable only over a VPN.** No services exposed on the public internet; agents do not hold SSH credentials between hosts. (Ch. 6, 10)
- [ ] **The harness exposes hooks at tool boundaries.** PreToolUse and PostToolUse hooks redact secrets, validate arguments, and emit audit events. (Ch. 6, 7)
- [ ] **Audit logs are written outside the agent's reach.** Tool calls, decisions, arguments (redacted), and outcomes stream to a tamper-evident store the agent cannot modify. (Ch. 6)
- [ ] **Resource and cost limits are configured.** Loop iteration cap, per-tool rate limit, token and cost budget, outbound bytes-per-session ceiling. (Ch. 6)

## Secrets, supply chain, channels, and memory

- [ ] **No cleartext secrets in workspaces or configuration files.** L1 floor; secrets injected via environment variables, ideally pulled from a password manager or vault at session start. (Ch. 7)
- [ ] **Where consequence is high, credentials are short-lived and just-in-time.** A vault issues per-call, scope-limited credentials; the agent's process does not see them. (Ch. 7)
- [ ] **Tool outputs and arguments pass through redaction hooks.** Known secret patterns stripped before content enters the model's context or leaves the harness. (Ch. 7)
- [ ] **Models, MCP servers, plugins, and skills are pinned to specific versions.** Auto-update is disabled; upgrades are deliberate. Tool descriptions inspected before deployment. (Ch. 8)
- [ ] **Channels are private and per-agent.** Bots in shared channels and unauthenticated email-to-agent paths are avoided unless explicitly required and confirmed. (Ch. 9)
- [ ] **Memory surfaces are reviewed as code.** Long-lived agent files (`CLAUDE.md` and equivalents) are committed, reviewed, and not auto-updated by the agent itself. (Ch. 9)

## Six threat vectors with one defence each

- **Prompt injection** → strict-ignore guard system prompt on every agent + provenance tagging on fetched content. (Ch. 4)
- **Rogue agent / over-privileged agent** → default-deny tools; narrow scopes per role; sandbox shell-using agents; no shared credentials. (Ch. 5)
- **Harness and runtime compromise** → default-deny outbound; VPN-gate everything internal; PreToolUse hooks; audit log outside the agent. (Ch. 6)
- **Secret leak** → secrets off disk to password manager or vault; redact in PostToolUse hooks; per-agent identity with scoped tokens. (Ch. 7)
- **Supply-chain compromise** → pin model and dependency versions; inspect tool descriptions; prefer signed, provenanced artefacts. (Ch. 8)
- **Channels, memory, multi-agent compromise** → private channels with per-agent identity; memory reviewed as code; inter-agent boundaries treated as external input. (Ch. 9)

## Adversarial review (run before sign-off)

- [ ] "What is the most damaging tool call that could be induced from a fetched webpage?" — verify controls hold.
- [ ] "What does the audit show after that tool call?" — verify log is reconstructable.
- [ ] "Which credential, if exfiltrated, lets the attacker pivot beyond this agent?" — verify scope and TTL.
- [ ] "If this agent is suspected compromised right now, what is the kill-switch command?" — verify revocation works.

## Sign-off

| Reviewer | Date | L-level summary | Notes |
|---|---|---|---|
| | | | |