Consultiply

• AI

MIP RESOURCE CENTER

Go-live is day one. Run every client agent on a written aftercare cadence.

A runbook for MSPs who run AI agents for clients after go-live. It covers the baseline to write on day one, weekly and monthly checks, model changes, the first hour of an incident and a clean handover.

This guide is Consultiply's independent reading of the cited publishers' pages. It is not affiliated with or endorsed by Microsoft, Huntress, ConnectWise, N-able, CRN, Channel Dive or Bain & Company, and it is not a recommendation to buy any product. Every figure is the named publisher's own.

Why go-live is the start of the work

Bain & Company's Technology Report 2026 (Sep 29, 2026) describes the agent ownership gap in large enterprises. Microsoft's Nov 12, 2026 research on coding agents supplies the model and logging checks. Huntress and Channel Dive cover a ScreenConnect incident, and an undated CRN page (read Nov 13, 2026) covers an N-able N-central emergency fix. Those are MSP tooling stories that did not involve AI agents. We borrow their lessons on leftover access, incident triage and vendor outages.

Bain says many companies "would struggle to say how many agents they have, who owns them, and what they are permitted to do". It also calls "installing a kill switch in each agent" a "crucial step".

For most agents, go-live is when they start acting on client systems with real credentials. After that, the register you wrote before launch drifts unless someone keeps it true. Two earlier Consultiply guides cover setup: the agent register and agent identity and inventory. This one covers the weeks after.

The cadence, checklists, owners and fields are Consultiply's suggestions, not sourced thresholds.

Day one: write the baseline

When an agent goes live, write one register entry per client:

- ☐ Named owner at your MSP and at the client
- ☐ Permissions: every account, role, API scope and shared key the agent holds
- ☐ Systems touched: every client system it can read from or write to
- ☐ Off switch: where it is, who can use it, the date you last tested it
- ☐ Model or checkpoint source, version and hash where one exists
- ☐ Tool list: every tool the agent can call, with its scope
- ☐ Fixed test set: a short list of real requests with expected output and tool calls
- ☐ Log location: where tool calls and actions are recorded, and for how long
- ☐ Weekly cost baseline from the first weeks of normal use

The model line comes from Microsoft's post on post-training supply chain attacks, written for teams running fine-tuned or third-party open-weight checkpoints: "Record checkpoint provenance: Capture the source, version, hash, available training disclosures, and review status."

On tools, Microsoft notes: "The available tools and their permissions define the potential blast radius."

The cadence at a glance

Change the owners to fit your team.

When	What	Owner of the check
Weekly	Logs, odd calls, approvals, cost, drift, vendor notices	Service desk lead
Monthly	Permissions, re-test, credential rotation, cost, client report	Service manager
On every model or version change	Record, re-test, stage, keep rollback	Engineer who owns the agent
On every incident	Off switch, out-of-band triage, evidence, client call	On-call engineer
At offboarding	Handover, revoke, confirm off switch, sign-off	Service manager with client owner

Weekly checks

Go through these one agent at a time. Quotes are Microsoft's:

- ☐ Confirm logging is on and complete: "Monitor tool activity: Log the selected tool, model-supplied parameters, resulting action, and outbound connections."
- ☐ Look for odd calls: "Investigate deviations: Alert on tool calls that do not follow from the user request or retrieved task context."
- ☐ Clear the approvals queue and note anything approved without a clear reason
- ☐ Compare cost with the weekly baseline and note any unexplained gap
- ☐ Check for drift: outputs or tool use that differ from the fixed test set
- ☐ Read vendor notices for the model, the agent platform and every tool it calls
- ☐ Contact the client owner if anything needs their decision

Monthly review

- ☐ Permissions review: remove tools, add-ons and shared keys the agent no longer needs. Microsoft: "Remove unnecessary tools and scope credentials, file paths, APIs, and network access to the agent's task."
- ☐ Re-run the fixed test set. Microsoft: "Evaluate the deployed configuration: Test repository content, tool results, trigger variations, held-out tasks, and the production harness schemas."
- ☐ Rotate the agent's credentials
- ☐ Review the month's cost against the baseline
- ☐ Send the client a plain-language report: what the agent did, what changed, what you removed, what needs their decision

On add-ons, Huntress reported from the ScreenConnect incident: "Our analysis suggests that most RMM+ installations we encountered were legitimate, usually added for a project and left in place." Huntress also wrote that "RMM+ bypasses ScreenConnect's user accounts, roles, and MFA, and authenticates requests with a shared access key stored in the extension's settings." Give each shared key an agent holds its own register line with an owner and rotation date.

Bain wants reports "granular, actionable, and at least monthly". That is a large-enterprise benchmark, and a sensible minimum for client reports too.

When the model or version changes

For every model, checkpoint or agent version change:

1. Record the new source, version and hash in the register before anything goes live.
2. Re-test in the deployed configuration with your fixed test set, not only on public benchmarks.
3. Stage the change on one client process or a test tenant first.
4. Keep the previous version ready to roll back to, and write down how.
5. Tell the client owner before the switch, with the date and the rollback plan.

Step 2 rests on Microsoft's warning: "Assume that a compromised model may behave normally on standard benchmarks", and the researchers "even observed improved performance on coding benchmarks, which could make compromised models more attractive to users."

Microsoft adds a caveat: "Because the backdoor lives in the model itself, these results don't indicate a weakness in the default Claude Code or Codex CLI configurations. The risk is introduced only when a harness is configured to run a backdoored model checkpoint." For hosted models, record the model name the provider shows and the switch date, and run the same test set.

Incident response: the first hour

These steps borrow Huntress and ConnectWise advice from the ScreenConnect incident, which did not involve an AI agent. The order is Consultiply's suggestion.

1. Stop the agent with the off switch. Note the time and who did it.
2. Do not investigate through the tool the agent or attacker used. Huntress advised "Do not triage infected hosts over ScreenConnect", and ConnectWise told partners to "Use separate tooling to investigate suspected compromise" (Channel Dive).
3. Secure admin access from a clean device, revoke sessions and rotate credentials. Huntress's version: "From a clean device, terminate active ScreenConnect web sessions and rotate affected ScreenConnect credentials". For an agent, that means every key and account in its register entry.
4. Preserve evidence. ConnectWise's list: "Preserve evidence, including authentication, extension installation, Run command and Toolbox records." For an agent, include its tool call records.
5. Tell the client owner what you stopped, when, and what happens next.
6. Review logs back to the date the vendor advises, or to the last clean baseline if there is none.

When your platform vendor goes dark

An RMM platform story, used by analogy. CRN reported that N-able had its hosted N-central environments offline for "roughly a day" after critical-severity vulnerabilities were publicly disclosed, and that "those operating their own N-central servers must apply the update themselves." Per CRN, N-able "has not disclosed that successful exploitation has occurred in customer environments", and Huntress said it had not observed it either. CRN added that "Huntress advised organizations to review login and authentication activity dating back to Nov. 8 for signs of unauthorized access."

Keep a written manual fallback for each client process an agent runs, in case the agent or its platform goes offline.

Offboarding and handover to the client

When a client leaves or takes the agent in-house:

- ☐ Hand over the register entry, logs, fixed test set and this runbook
- ☐ Confirm a named client owner in writing
- ☐ Revoke every credential and shared key you issued or hold
- ☐ Remove the agent from every tool and integration on your side
- ☐ Test the off switch with the client watching before you leave
- ☐ Get written sign-off on the handover
- ☐ Send a final cost statement

The numbers, and what they are

Finding	Figure	Publisher
Managed endpoints with rogue ScreenConnect clients	At least 288 endpoints across 48 organizations (Huntress's figure)	Huntress
Partner environments where ConnectWise disabled five extensions	More than 400 (Huntress's figure)	Huntress
Time hosted N-central was offline	Roughly a day (CRN's report)	CRN
Share of training corpus poisoned in the test	About 1%, 90 of roughly 9,000 conversations (Microsoft's experiment)	Microsoft

Read with care

- Microsoft tested poisoned datasets and checkpoints on coding agents, not MSP deployments.
- ConnectWise said "the initial access method remains unconfirmed" (Channel Dive).
- Channel Dive reports about 400 partner environments; Huntress reports more than 400.

> This guide is Consultiply's independent reading of the cited publishers' pages. It is not affiliated with or endorsed by Microsoft, Huntress, ConnectWise, N-able, CRN, Channel Dive or Bain & Company, and it is not a recommendation to buy any product. Every figure is the named publisher's own.

Reach us at transform@consultiply.ai

Sources

Trained to betray: How post-training supply chain attacks hijack coding agents (Microsoft Security Blog, Microsoft Security Research and Microsoft AI Red Team, Nov 12, 2026)

SCKit Worm Installs Rogue ScreenConnect Clients and Session Cookie Relays Across Hundreds of Endpoints (Huntress, Nov 11, 2026)

ConnectWise disables third-party extensions, probes ScreenConnect cyberattack (Channel Dive, first published on Cybersecurity Dive, David Jones, Nov 12, 2026)

N-able N-central 'Able2Shell' Vulnerabilities: 5 Things To Know (CRN, Kyle Alspach, no date shown on page, read Nov 13, 2026)

Cybersecurity's New AI Imperative: Attacking the Backlog of Vulnerability Alerts (Bain & Company, Technology Report 2026, Sep 29, 2026, pp. 45 and 48)
