

Three real n8n agent changes, reviewed before handoff

I ran ReleaseGuard on three consecutive changes to a public, MIT-licensed n8n alert triage template and then read every result by hand. This case shows the tool output and my notes side by side, including where the tool was wrong.

Source	https://github.com/scanner-inc/agents , n8n/alert-triage/workflow.json
Changes reviewed	A: Three threat intel tools are added to the agent B: Optional Jira ticket creation, shipped disabled C: The agent iteration limit goes from 10 to 100
Engine	ReleaseGuard 1.3.0, 2026-09-30

I reviewed every result and signed off the notes on 30 September 2026. This is an independent review of a public MIT template. It makes no claim about Scanner's own systems and it is not a penetration test.

1 Source and license

The workflow is `n8n/alert-triage/workflow.json` in the public `scanner-inc/agents` repository. A webhook receives a detection alert, an AI agent investigates it with read tools, and the result is posted to Slack. I compared the file at consecutive commits. I did not run the workflow, call any API or look at any Scanner system. This is a review of a public template only; it says nothing about how Scanner runs its own product.

The workflow files are Copyright (c) 2026 Scanner, Inc. and released under the MIT License. This review is independent and is not affiliated with or endorsed by Scanner. The raw workflow files are not republished here; the commit links and hashes below point to the exact versions.

Each export carries a sample alert in its pinned data, with a tenant ID and a link into the Scanner app. I left both out of this page, the receipts and the report.

Open the repository	https://github.com/scanner-inc/agents
Open the MIT license	https://github.com/scanner-inc/agents/blob/15a608933b52ae2acc43777d5a46d9e1fa795e92/LICENSE

What this case does not show

- It is a static comparison of exported JSON. I did not run the workflow, the model or any of the lookup APIs.
- It does not show which tools the Scanner MCP server offers. The export attaches the MCP client with its default tool selection, so the agent gets whatever the server exposes.
- Production credentials, n8n instance settings and the sending side of the alerts were not visible to me.
- It is not a penetration test, a vulnerability verdict about Scanner, a certificate or a security guarantee.

2 Change A: Three threat intel tools are added to the agent

What changed

Three HTTP tools are attached to Alert Triage Agent. ThreatFox IOC Lookup sends a POST to threatfox-api.abuse.ch with the query search_ioc and an IOC the model fills in. OTX Pulse Search sends a GET to otx.alienvault.com with a keyword the model picks. Feodo Tracker downloads the Feodo Tracker IP blocklist and takes no input.

The system prompt gets a short section that tells the agent when to use the three tools. The webhook, the Slack step and the model stay the same.

Before	e425e65, 2026-04-23, "Rename Scanner bearer credential to Scanner API/MCP Bearer Auth account across all workflows" https://github.com/scanner-inc/agents/commit/e425e65c11cf64f5fc39e2f52027c490c66efe79
After	03ccb47, 2026-04-23, "Add threat intel tools (ThreatFox, OTX, Feodo) to alert-triage and slack-bot" https://github.com/scanner-inc/agents/commit/03ccb47972ca8b364b3ed8698a7e825da134defb

Tool output

ReleaseGuard 1.3.0, exported JSON only, no staging run.

Result	Human validation complete. Release decision required
Records	0 blockers, 6 to review, 0 improvements, 3 context
Open findings	1 from this change, 6 already present before
Same pair before the fixes (engine 1.2.0)	Stop this change. The one blocker was TG-106: the ThreatFox lookup is a POST, and the old engine counted every POST as a write. (1 blocker, 5 to review, 0 improvements, 3 context)
CR-001, REVIEW	A new medium finding was introduced: Outbound requests do not show an explicit timeout or retry policy
CR-002, REVIEW	A new external destination appears: feodotracker.abuse.ch
CR-003, REVIEW	A new external destination appears: otx.alienvault.com
CR-004, REVIEW	A new external destination appears: threatfox-api.abuse.ch
CR-005, REVIEW	A POST request is treated as a read-only lookup: ThreatFox IOC Lookup
CR-006, REVIEW	Security-relevant configuration changed: Alert Triage Agent

Confirmed

- CR-001, no timeout or retry on the three new tools (AA-008). A slow or unreachable lookup API holds the agent run, and the agent node itself retries up to 3 times.
- CR-002 to CR-004, three new outside destinations: abuse.ch for ThreatFox and Feodo Tracker, and AlienVault OTX. What leaves the workflow is the value the model chooses. For ThreatFox that is one IOC. For OTX Pulse Search it is any keyword; the prompt suggests IPs, domains, CVE IDs, malware and actor names, but nothing stops the model from sending an internal host name or a user name taken from the alert.
- CR-005, ThreatFox IOC Lookup is a read-only lookup. The body asks for search_ioc; I checked that no write verb is in the body.

False positives

- Before the fix, TG-106 was a blocker because the POST to ThreatFox was counted as a write. The request is a search. The tool now records it as a lookup to review instead of a blocker.

What the tool missed

- The OTX keyword goes into the URL query string as the model wrote it (q=...). There is no encoding step, so the model's text becomes part of the URL. The host stays fixed, so the impact is low.
- Feodo Tracker returns the whole blocklist to the model on every call, as its tool description says. That costs context and money on each run. I did not measure the list size.

Recommended fix before handoff

- Set a timeout of about 10 to 15 seconds and one retry on the three tools.
- Limit OTX to lookups of a given indicator type. The next commit on this file (4abe04e, 24 April 2026) did this: it replaced OTX Pulse Search with OTX IOC Lookup and removed Feodo Tracker.
- Write in the handoff note which values go to abuse.ch and AlienVault, so the client can decide whether that is acceptable.

3 Change B: Optional Jira ticket creation, shipped disabled

What changed

The system prompt now asks the agent to end every answer with a Jira block after a `===JIRA===` marker. The model decides `create` true or false with a rule written in the prompt: `SUSPICIOUS` or `MALICIOUS`, and the alert severity `Medium` or higher.

A new Code step, `Split Output`, splits the answer into the Slack text and the Jira fields. Send a message now posts that Slack text. A new If step, `Create Jira?`, checks `create_jira`. A new Jira step, `Create Jira Issue`, creates an issue with the summary, description, priority and labels the model wrote. It ships disabled; the commit message says the workflow runs Slack only out of the box.

Before	cfcaea6, 2026-04-24, "Tighten alert-triage report formatting" https://github.com/scanner-inc/agents/commit/cfcaea60035aebf3cd9b2e324dc7d3a901f9b0ce
After	15a6089, 2026-05-12, "Add optional Jira ticket creation to n8n alert-triage and threat-hunt" https://github.com/scanner-inc/agents/commit/15a608933b52ae2acc43777d5a46d9e1fa795e92

Tool output

ReleaseGuard 1.3.0, exported JSON only, no staging run.

Result	Human validation complete. Release decision required
Records	0 blockers, 3 to review, 0 improvements, 5 context
Open findings	0 from this change, 7 already present before
Same pair before the fixes (engine 1.2.0)	Stop this change. The two blockers were TG-100 and AA-004 reported as new, and the same two rules were also reported as resolved. Only a step on their path had changed. The disabled Jira step appeared only as context. (2 blockers, 2 to review, 2 improvements, 3 context)
CR-001, REVIEW	A write path was added but ships disabled: Create Jira Issue
CR-002, REVIEW	Security-relevant configuration changed: Alert Triage Agent
CR-003, REVIEW	Security-relevant configuration changed: Send a message
CR-004, CONTEXT	An existing finding now covers different steps: A prompt-injection source can steer a privileged action through the AI agent
CR-005, CONTEXT	An existing finding now covers different steps: A model can reach an external write action without an approval step

Confirmed

- CR-001, a write path that ships disabled, behind a gate that decides on model output. Whether a ticket is opened, and every field in it, comes from the model. Split Output checks that the Jira block parses as JSON and that create is true, and falls back to Slack only; that is a real guard against broken output. It does not check the classification or the severity against the alert itself. Turning the Jira step on is one toggle in the editor.
- CR-003, Send a message changed. The old expression cut everything before the siren emoji; the new one posts the Slack text as it is, so any preamble the model writes now reaches the channel. The maintainers restored the cut the same day in a1dd1a6, which confirms the regression.

False positives

- Before the fix, TG-100 and AA-004 were reported as new blockers and at the same time as resolved. Both still start at the webhook and end at Send a message; only Split Output was added in between. The tool now records one context line for each.

What the tool missed

- Labels and priority go to Jira exactly as the model wrote them. Jira rejects some label characters, and some projects reject a priority. The maintainers fixed both in a1dd1a6 the same day. The tool cannot know Jira's field rules.
- Nothing limits the number of tickets. A burst of alerts can open one ticket per alert, and there is no check that one alert opens only one ticket.
- The severity threshold is a sentence in the prompt, not a check in the workflow.

Recommended fix before handoff

- Before anyone enables Create Jira Issue: compute create_jira in Split Output from the alert's own severity field (read from the Webhook item) and from a classification that must be exactly SUSPICIOUS or MALICIOUS.
- Keep labels to a fixed list, use the alert ID as a dedupe key, and set a daily ticket cap.
- Treat switching the Jira step on as its own change, reviewed on its own.

4 Change C: The agent iteration limit goes from 10 to 100

What changed

One field changes. Alert Triage Agent gets options.maxIterations set to 100. Before, the field was not set, so n8n's default of 10 applied. The commit message gives the reason: with 10, agents stopped in the middle of an investigation.

Before	4abe04e, 2026-04-24, "Modernize n8n tool nodes and refactor OTX threat intel" https://github.com/scanner-inc/agents/commit/4abe04ef4057778425cd0200e657b7ac71db10f8
After	9d6164e, 2026-04-24, "Bump n8n agent maxIterations to 100" https://github.com/scanner-inc/agents/commit/9d6164e147d540cbdfed1abdb43fac18a34c58df

Tool output

ReleaseGuard 1.3.0, exported JSON only, no staging run.

Result	Human validation complete. Release decision required
Records	0 blockers, 1 to review, 0 improvements, 0 context
Open findings	0 from this change, 7 already present before
Same pair before the fixes (engine 1.2.0)	Stop this change, with no blocker in the change itself. The only record was a generic configuration change; the stop came from findings the template already had. (0 blockers, 1 to review, 0 improvements, 0 context)

CR-001, REVIEW	An agent limit was loosened: Alert Triage Agent
----------------	---

REVIEWER NOTES, AHMET GÖKER, 30 SEPTEMBER 2026

Confirmed

- CR-001, the agent limit was loosened from 10 (n8n default) to 100. The reason in the commit is sound. What it adds: one alert can now drive up to 100 agent steps, each a model call and often a tool call. The agent step also retries on failure up to 3 times with a 5 second wait, so the worst case per alert is about 300 steps. The workflow settings set no execution timeout, and no rate limit sits before the agent (AA-006, already present).

False positives

- None in the new output. Before the fix, the result was Stop this change for a one-field change whose risk is cost and run time, because findings the template already had decided the result.

What the tool missed

- The tool reports the new limit, but it does not multiply it by the agent's retries or check for an execution timeout in the workflow settings. I added both by hand.

Recommended fix before handoff

- Keep 100 if investigations need it, and add a limit around it: an execution timeout in the workflow settings, one retry at most on the agent step, and a cap or dedupe on alerts before the agent.
- Watch model spend per run for the first week after the change.

5 Findings the template already had

These open findings were already in the version before each change. AA-008 counts as already present from change B on, because change A brought it in. They are listed in every receipt, but they do not decide any of the three changes.

REVIEWER NOTES, AHMET GÖKER, 30 SEPTEMBER 2026

CRITICAL TG-100	A prompt-injection source can steer a privileged action through the AI agent The path from Webhook through Alert Triage Agent to Send a message is real: the whole alert JSON goes into the prompt, and alert fields can hold text an attacker controls, such as a user or resource name. The webhook uses header auth and the end step posts to one fixed Slack channel, so the worst case is a wrong or misleading triage note. I would rate it medium for this template, not critical.
HIGH AA-003	Untrusted input can reach a model without a visible validation boundary Confirmed. The prompt is built with JSON.stringify of the whole request item, with no step that limits size or strips free text first. For a triage agent this is by design, which is why the tools it can call should stay read-only.
HIGH AA-004	A model can reach an external write action without an approval step Confirmed and by design: the model output is posted to Slack with no approval step. Posting the triage note is the purpose of the workflow. Approval is not needed here, but it is needed before any step that writes somewhere else.
HIGH AA-006	No rate or cost boundary was detected before model usage No rate or cost limit is visible in the export. The alerts come from the detection platform behind header auth, so a limit may exist on the sending side; the export cannot show it. Ask the team where the alert rate is capped.
MEDIUM AA-005	No structured model-output validation was detected Confirmed. The Slack text is the model output with only a cut at the first siren emoji. No schema check runs on it.
MEDIUM AA-008	Outbound requests do not show an explicit timeout or retry policy Confirmed. The HTTP tools set no timeout and no retry, so a slow lookup API holds the whole agent run.
LOW AA-010	No workflow-level failure route was detected Confirmed, low. No error workflow is set, so a failed triage run goes unnoticed unless someone looks at the execution list.