

Quick Setup Guide — Wazuh → n8n Alert Workflow

This short documentation explains, step by step, how to wire Wazuh alerts into an n8n workflow and enrich them with an AI-generated HTML summary and recommendations. Screenshots will be added later.

1) Prerequisites

- A running **Wazuh Manager** with shell access.
 - An **n8n** instance you can reach from the Wazuh Manager (HTTP/HTTPS).
 - The four files from this repository:
 - `custom-n8n` (Bash)
 - `custom-n8n.py` (Python)
 - `AI-Agent-Prompt.txt`
 - `ossec.conf.txt`
-

2) Install the Wazuh integration scripts

Copy the two helper scripts to Wazuh's integrations directory, set ownership, and make them executable:

```
sudo cp custom-n8n /var/ossec/integrations/
sudo cp custom-n8n.py /var/ossec/integrations/

sudo chown root:wazuh /var/ossec/integrations/custom-n8n
sudo chown root:wazuh /var/ossec/integrations/custom-n8n.py

sudo chmod +x /var/ossec/integrations/custom-n8n
sudo chmod +x /var/ossec/integrations/custom-n8n.py
```

These scripts are invoked by Wazuh when an alert matches your configuration.

3) Adjust the Wazuh configuration

1. Open the manager configuration:
2. `sudo nano /var/ossec/etc/ossec.conf`
3. Insert the configuration block from **`ossec.conf.txt`** into the correct section (inside `<ossec_config> ... </ossec_config>`).

The snippet in `ossec.conf.txt` enables the webhook flow by calling `custom-n8n` on new alerts.

4. Validate the XML (no duplicate tags, proper nesting).
 5. Restart the Wazuh Manager:
 6. `sudo systemctl restart wazuh-manager`
 7. Check logs to confirm the manager started cleanly:
 8. `sudo tail -f /var/ossec/logs/ossec.log`
-

4) Create the n8n workflow (minimal version)

1. **Webhook Trigger**
 - Add a **Webhook** node.
 - Method: `POST`
 - Path: for example `/wazuh/alert`
 - Copy the Production URL — you'll reference it in the Wazuh integration snippet (already covered by `ossec.conf.txt`).
 2. **Normalize/Select fields (optional but recommended)**
 - Add a **Function** (Code) node to pick the most important fields (rule, level, agent, timestamp, source IP, file, etc.).
 - Output a compact JSON object for the AI prompt.
 3. **AI Summary**
 - Add your preferred LLM node (e.g., OpenAI).
 - Paste the content of **AI-Agent-Prompt.txt** as the prompt.
 - Provide the normalized alert JSON as context/variables so the AI can generate an **HTML** summary and **actionable recommendations**.
 4. **Delivery**
 - Choose how to consume the HTML:
 - Return it to the Webhook response,
 - Send via Email/Slack/Teams,
 - Store it (Database, Notion, etc.).
 5. **Response**
 - Ensure the workflow returns an HTTP `200` to the Webhook with a small JSON/HTML confirmation.
-

5) Test the end-to-end flow

- **From n8n:** Send a sample `POST` to the Webhook URL with a minimal JSON body to verify connectivity.
 - **From Wazuh:** Trigger a test alert (for example, using `wazuh-logtest`) and watch:
 - `/var/ossec/logs/ossec.log` on the manager,
 - The **Executions** list in n8n,
 - Your chosen delivery channel (email/message).
-

6) Troubleshooting

- **Permissions:** Confirm both scripts exist, are owned by `root:wazuh`, and are executable.

- **Paths:** The integration path must be exactly `/var/ossec/integrations/`.
 - **Config:** Re-check the block inserted from `ossec.conf.txt` and that it sits within `<ossec_config>`.
 - **Connectivity:** The Wazuh host must reach the n8n Webhook URL (firewall/SSL/proxy).
 - **Logs:**
 - Wazuh: `/var/ossec/logs/ossec.log`
 - n8n: Instance logs and the execution detail view.
-

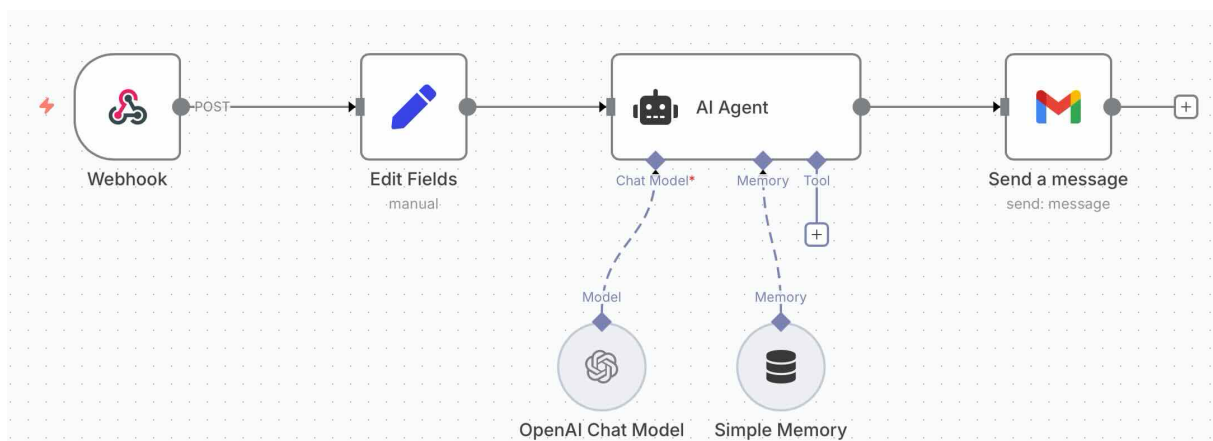
7) Security Notes

- Limit access to the n8n Webhook (IP allowlist, secret token, or auth).
 - Avoid sending sensitive data unencrypted; prefer HTTPS.
 - Keep secrets (API keys, tokens) in n8n credentials, not in node parameters or prompts.
-

8) What each file is for (quick recap)

- **custom-n8n** — Bash script called by Wazuh to forward alerts.
 - **custom-n8n.py** — Python helper for pre-processing before sending.
 - **AI-Agent-Prompt.txt** — Prompt template that instructs the AI to produce an HTML summary and remediation suggestions.
 - **ossec.conf.txt** — Config snippet you paste into `/var/ossec/etc/ossec.conf` to enable the integration.
-

Screenshots:



Workflow

Webhook Listen for test event

Parameters Settings Docs

Webhook URLs

Test URL Production URL

POST `https://b6b1-home-8a5/webhook-test/d9f7d5e6-640e-4743-bf36-d4170ebe7946`

HTTP Method: POST

Path: `/webhook-test/d9f7d5e6-640e-4743-bf36-d4170ebe7946`

Authentication: None

Respond: Immediately

If you are sending back a response, add a "Content-Type" response header with the appropriate value to avoid unexpected behavior

Options: No properties

Add option

Webhook Trigger

Edit Fields Execute step

Parameters Settings Doc

Mode: Manual Mapping

Fields to Set

body.alert.full_log
String
= `{{ $json.body.alert.full_log }}`

body.alert.predecoder
Object
= `{{ $json.body.alert.predecoder }}`

body.fields
Object
= `{{ $json.body.fields }}`
Fixed Expression

Drag input fields here or Add Field

Include Other Input Fields: ☐

Options: No properties

Add option

Select fields

AI Agent
Execute step

Parameters
Settings
Doc

Get started faster with our [pre-built agents](#)

Source for Prompt (User Message)

Define below

Prompt (User Message)

fx

Du bist ein erfahrener SOC-Analyst. Analysiere den folgenden Wazuh-Alarm und ERZEUGE als Ausgabe ausschließlich ein `<div>...</div>`

Require Specific Output Format

Enable Fallback Model

Options

No properties

Add Option

AI Summary

OpenAI Chat Model

Parameters
Settings
Doc

Credential to connect with

OpenAi account

Model

From list

chatgpt-4o-lat...

Options

No properties

Add Option

OpenAI Chat Model

Simple Memory

Parameters
Settings
Doc

Session ID

Define below

Key

my-n8n-id

Context Window Length


5

How many past interactions the model receives as context

Simple Memory

Sort your Gmail inbox using our pre-built [Email triage agent](#)

Credential to connect with

Gmail account 

Resource

Message

Operation

Send

To

test@gmail.de

Subject

Wazuh Alarm

Email Type

HTML

Message

fx

{{ \$json.output }}

Options

Append n8n Attribution

☐

Add option

Send an E-Mail