// KI-Hub VON STRUKTURAFLOW

IT-SECURITY

n8n Workflow auf Sicherheitslücken testen: KMU-Guide

n8n-Workflows absichern: Webhook-Tests, Credential-Audit und DSGVO-Checkliste — praxisnah für KMU ohne eigenes Security-Team.

AUTOR
Strukturaflow-Team

VERÖFFENTLICHT
31. Mai 2026

ONLINE LESEN
https://wissen.strukturaflow.it.com/n8n-workflow-sicherheitsluecken-testen/

Ein Webhook-Trigger ohne Authentifizierung, ein Formular das Nutzer-Input direkt in eine HTTP-Anfrage übergibt, ein API-Key der seit zwei Jahren nicht rotiert wurde — und plötzlich wird Ihre Kundendatenbank exportiert. Niemand bemerkt es wochenlang, weil der Workflow einfach weiterläuft. Das ist kein theoretisches Schreckensszenario, sondern ein realistisches Angriffsmuster, das wir in der Praxis regelmäßig antreffen.

n8n ist ein leistungsfähiges Automatisierungswerkzeug — aber Sicherheit ist kein Standardzustand, weder bei der Self-hosted-Variante noch in der Cloud. KMU, die personenbezogene Daten durch Workflows leiten, tragen die volle DSGVO-Verantwortung für diese Datenpfade.

Dieser Artikel liefert einen strukturierten Testprozess in fünf Schritten, eine kopierbare DSGVO-Checkliste und konkrete Werkzeuge — umsetzbar auch ohne dediziertes Security-Team.

Warum n8n-Workflows ein unterschätztes Angriffsziel sind

n8n sitzt als Integrationsschicht zwischen vielen Systemen gleichzeitig: CRM, E-Mail-Dienst, Datenbank, externe APIs. Das bedeutet eine breite Angriffsfläche — wer Zugang zum n8n-Workflow gewinnt, hat oft Zugang zu allem, was dieser Workflow berührt.

Die typische Fehleinschätzung lautet: „Es ist ja nur Automatisierung." Tatsächlich laufen n8n-Instanzen häufig mit Admin-Credentials für mehrere Systeme. Ein kompromittierter Workflow ist damit kein isoliertes Problem, sondern ein Einfallstor in die gesamte Systemlandschaft.

Welche Angriffsvektoren sind bei n8n konkret realistisch?

- **Webhook-Missbrauch:** Öffentlich erreichbare Webhook-Endpunkte ohne Authentifizierung können von jedem aufgerufen werden — auch von Skripten, die automatisiert Daten extrahieren oder den Workflow gezielt mit schadhaften Payloads fluten.
- **Credential-Leakage:** Hartcodierte API-Keys in Workflow-Expressions oder Execution-Logs, die personenbezogene Daten enthalten und nicht bereinigt werden.
- **Server-Side Request Forgery (SSRF) via HTTP-Node:** Wenn externe Eingaben die Ziel-URL eines HTTP-Request-Nodes beeinflussen können, lassen sich intern erreichbare Dienste ansprechen.
- **Unsichere Umgebungsvariablen:** Fehlkonfigurierte `.env`-Dateien auf Self-hosted-Instanzen, die Secrets im Klartext exponieren.

DSGVO Art. 32 Abs. 1 lit. b und d verpflichtet Unternehmen zu technischen Schutzmaßnahmen — dazu zählen ausdrücklich die Sicherstellung der Vertraulichkeit und ein Verfahren zur regelmäßigen Überprüfung der Sicherheitsmaßnahmen. Das gilt auch für Automations-Workflows. Mehr zum Thema KI-Sicherheit im Unternehmenskontext finden Sie im Artikel KI-Workflows mit Unternehmensdaten absichern.

Self-hosted vs. n8n Cloud – wo liegen die Sicherheitsunterschiede?

Die Wahl zwischen Self-hosted und n8n Cloud verschiebt die Verantwortlichkeiten — sie hebt sie nicht auf.

Self-hosted: Sie haben volle Kontrolle über Netzwerkkonfiguration, TLS-Zertifikate, Update-Zyklen und Zugriffsschutz. Das bedeutet auch: Sie tragen die volle Verantwortung. Wer keine dedizierten IT-Ressourcen hat, unterschätzt den operativen Aufwand leicht.

n8n Cloud: Die Infrastruktur-Sicherheit liegt bei n8n.io. Netzwerk, Patching und Verfügbarkeit sind Sache des Anbieters. Aber: Workflow-Logik, API-Keys und Datenpfade bleiben vollständig in Ihrer Verantwortung.

DSGVO-Kontext: Bei Self-hosted-Betrieb in Deutschland, Österreich oder der Schweiz müssen Sie den Serverstandort dokumentieren und einen Auftragsverarbeitungsvertrag (AVV) mit dem Hosting-Anbieter abschließen. Bei n8n Cloud ist zu prüfen, ob und wo Daten verarbeitet werden und ob ein AVV mit n8n.io vorhanden ist.

Eine grobe Orientierungshilfe:

	WENIG IT-RESSOURCEN	STARKE IT-RESSOURCEN
Niedrige Datensensibilität	n8n Cloud	Self-hosted oder Cloud
Hohe Datensensibilität	n8n Cloud + AVV + externer Review	Self-hosted mit vollständigem Sicherheitsmanagement

Ein Randthema, das in der Praxis unterschätzt wird: Wenn Mitarbeitende eigene n8n-Instanzen aufsetzen — etwa für interne Abteilungs-Automationen — entsteht ein klassisches Shadow-IT-Szenario mit unkontrollierten Datenpfaden. Das Thema wird im Artikel zu unkontrollierten KI-Tools im Unternehmen ausführlicher behandelt.

Schritt 1 – Bestandsaufnahme: Was läuft überhaupt durch Ihre Workflows?

Bevor Sie irgendetwas testen, brauchen Sie einen vollständigen Überblick. Ohne Inventar kein gezielter Test.

Workflows inventarisieren: Über die n8n API lässt sich die vollständige Workflow-Liste exportieren (`GET /workflows`). Prüfen Sie, welche Workflows aktiv sind — inaktive Workflows sind kein totes Gewicht, sie können reaktiviert werden und enthalten oft alte Credentials.

Datenklassen identifizieren: Welche Workflows verarbeiten personenbezogene Daten, Zahlungsinformationen oder interne Zugangsdaten? Diese Kategorisierung ist die Basis für jede weitere Risikoabwägung — und für Ihr Verarbeitungsverzeichnis nach DSGVO Art. 30.

Credentials-Audit: n8n speichert Credentials verschlüsselt im Credential Store. Der Encryption Key selbst muss aber separat gesichert sein — verlieren Sie ihn, sind alle Credentials unbrauchbar; landet er in falschen Händen, ist der Schutz hinfällig. Prüfen Sie: Wer hat überhaupt Zugriff auf den n8n-Credential Store? Sind Berechtigungen nach dem Least-Privilege-Prinzip vergeben?

Bestandsaufnahme-Checkliste:

- [ ] Alle aktiven Workflows dokumentiert und versioniert?
- [ ] Alle Workflows auf verarbeitete Datenklassen geprüft?
- [ ] Credentials auf minimale Berechtigungen (Least Privilege) geprüft?
- [ ] Inaktive Workflows deaktiviert oder gelöscht?
- [ ] n8n Encryption Key separat und sicher gespeichert?
- [ ] Zugriffsliste auf n8n-Instanz aktuell und bereinigt?

Wenn Sie KI-Agenten über n8n anbinden, ist die Frage des Datenzugriffs noch drängender — dazu liefert der Artikel KI-Agenten Datenzugriff kontrollieren: KMU-Guide weiterführende Hinweise.

Schritt 2 – Webhook-Sicherheit testen

Webhooks sind der häufigste Einstiegspunkt für Angriffe auf n8n-Instanzen. Ein Webhook ohne Authentifizierung ist technisch gesehen ein offenes Formular, das jeder im Internet absenden kann.

Testszenarien, die Sie manuell durchführen können:

Test 1 — Webhook ohne Auth aufrufen:

```
curl -X POST https://ihre-n8n-instanz.example.com/webhook/ihr-webhook-pfad \
  -H „Content-Type: application/json“ \
  -d '{„test“: „payload“}'
```

Wenn der Workflow ausgelöst wird, ohne dass Sie einen Auth-Header mitgegeben haben: Der Webhook ist offen. Das ist das häufigste Problem, das wir bei der Erstprüfung von n8n-Instanzen sehen.

Test 2 — Payload-Injection:

Was passiert bei unerwarteten JSON-Feldern oder überlangen Strings? Schicken Sie Payloads mit Sonderzeichen, verschachtelten Objekten oder Strings mit 10.000 Zeichen. Beobachten Sie, ob der Workflow damit umgehen kann oder ob er in einen Fehlerzustand gerät, der Informationen über die interne Systemstruktur preisgibt.

Test 3 — Rate-Limiting:

Kann ein Angreifer den Webhook per Skript fluten? Führen Sie 50–100 Anfragen in kurzer Folge durch und prüfen Sie, ob es eine Begrenzung gibt. Ohne Rate-Limiting riskieren Sie sowohl Ressourcen-Überlastung als auch Datenmissbrauch durch Massenabfragen.

Konkrete Absicherungsmaßnahmen:

- **Header-Auth:** In den Webhook-Node-Einstellungen unter „Authentication“ auf „Header Auth“ wechseln und einen eigenen Header-Key mit Secret definieren.
- **HMAC-Signaturprüfung:** Für Webhooks von Drittanbietern (z. B. Stripe, GitHub) immer die mitgelieferte Signatur verifizieren.
- **IP-Whitelisting:** Wenn der Webhook nur von bekannten Quellen aufgerufen werden soll, den Zugriff auf IP-Ebene einschränken — über Reverse-Proxy oder Firewall-Regel.

Beispiel – Webhook mit HMAC-Signatur absichern

Ein eingehender Webhook-Request bringt einen Header `X-Signature: sha256=<hash>` mit. In einem Code-Node direkt nach dem Webhook-Trigger prüfen Sie die Signatur gegen den bekannten Secret-Wert:

```
const crypto = require('crypto');
const secret = $env.WEBHOOK_SECRET;
const payload = JSON.stringify($input.first().json);
const signature = $input.first().headers['x-signature'];

const expectedSignature = 'sha256=' + crypto
  .createHmac('sha256', secret)
  .update(payload)
  .digest('hex');

if (signature !== expectedSignature) {
  throw new Error('Ungültige Signatur – Anfrage abgelehnt');
}

return $input.all();
```

Der `WEBHOOK_SECRET` wird dabei als Umgebungsvariable übergeben, nicht hart im Code kodiert.

Schritt 3 – HTTP-Request-Nodes und SSRF-Risiken prüfen

Server-Side Request Forgery (SSRF) ist ein Angriffstyp, bei dem ein Angreifer das System dazu bringt, Anfragen an interne Dienste zu stellen — Dienste, die von außen eigentlich nicht erreichbar sind.

In n8n entsteht dieses Risiko, wenn externe Eingaben (z. B. aus einem Webformular oder einer API-Antwort) die Ziel-URL eines HTTP-Request-Nodes beeinflussen können. Beispiel: Ein Workflow nimmt eine URL aus einem Formularfeld entgegen und ruft diese direkt auf. Ein Angreifer gibt `http://169.254.169.254/latest/meta-data/` ein — die AWS-Metadata-Adresse. Auf einer Cloud-Instanz könnte das sensible Infrastrukturinformationen zurückliefern.

Test: Durchsuchen Sie Ihre Workflows nach HTTP-Request-Nodes, deren URL-Feld einen Expression-Ausdruck enthält, der auf externe Inputs zurückgreift. Das Muster sieht etwa so aus: `{{ $json.url }}` oder `{{ $node[„Webhook“].json[„target“] }}`.

Absicherung:

- URL-Validierung im Code-Node vor dem HTTP-Request: Prüfen Sie Schema (nur `https://`), Domain gegen eine Allowlist erlaubter Domains, und blockieren Sie private IP-Ranges.
- Allowlisting: Definieren Sie explizit, welche Domains ein Workflow ansprechen darf — alles andere wird abgelehnt.

```
const allowedDomains = ['api.ihrcrmtool.com', 'hooks.slack.com'];
const url = new URL($json.targetUrl);

if (!allowedDomains.includes(url.hostname)) {
  throw new Error(`Domain nicht erlaubt: ${url.hostname}`);
}
```

SSRF-Risiken sind besonders relevant bei Workflows, die Nutzer-Input direkt weiterverarbeiten — etwa Kontaktformulare, die Daten an externe Dienste weiterleiten.

Schritt 4 – Credentials und Umgebungsvariablen absichern

API-Keys, Passwörter und OAuth-Token gehören ausnahmslos in den n8n Credential Store oder in Umgebungsvariablen — niemals direkt in Workflow-Expressions oder Code-Nodes.

Warum das in der Praxis trotzdem passiert: Beim schnellen Testen fügt man einen Token direkt in eine Expression ein. Der Workflow funktioniert, das Testen wird vergessen — und der Key steht fortan im Workflow-JSON, das ggf. mit Kollegen geteilt oder in einem Repository versioniert wird.

Test — Hartcodierte Secrets im Workflow-Export finden:

Exportieren Sie alle Workflows als JSON (`GET /workflows` oder über die n8n UI) und suchen Sie nach gängigen Mustern:

```
grep -E „(Bearer |api_key|secret|password|token)“ workflow-export.json
```

Oder spezifischer nach Base64-kodierten Strings oder typischen API-Key-Präfixen (z. B. `sk-` , `xoxb-` , `AIza`).

Umgebungsvariablen vs. Credential Store:

- **Credential Store:** Für Dienst-Credentials (OAuth, API-Keys zu externen Diensten) der empfohlene Weg in n8n. Credentials sind verschlüsselt gespeichert und nur innerhalb von Workflows verwendbar.
- **Umgebungsvariablen (.env):** Für globale Konfigurationswerte (z. B. den WEBHOOK_SECRET von oben). Vorteil: Kein Risiko durch Workflow-Export. Nachteil: Bei Self-hosted muss die `.env` -Datei selbst gesichert und aus der Versionskontrolle ausgeschlossen sein (`.gitignore`).

Rotationsplan: Wann wurden Ihre n8n-Credentials zuletzt rotiert? Bei Personalwechsel mit n8n-Zugang ist sofortige Rotation Pflicht. Empfehlenswert: Einen festen Rotationszyklus für kritische Credentials (z. B. alle 90 Tage) im Kalender verankern.

Der DSGVO-Bezug: Zugangsdaten zu Systemen mit personenbezogenen Daten unterliegen nach Art. 32 DSGVO besonderem Schutzbedarf. Eine fehlende Rotation ist im Fall eines Datenlecks ein dokumentierbarer Mangel an technischen Schutzmaßnahmen.

Schritt 5 – Automatisierte Sicherheitsprüfung einrichten

Manuelle Tests sind ein guter Startpunkt, aber kein dauerhaftes Sicherheitskonzept. Für KMU ohne Security-Team stellt sich die Frage: Was lässt sich mit vertretbarem Aufwand automatisieren?

Option A — n8n-eigener Sicherheits-Monitor:

Ein einfacher Monitoring-Workflow läuft täglich und prüft: - Wurden neue Webhooks aktiviert? (Vergleich der Webhook-Liste gegen eine bekannte Baseline) - Gibt es Execution-Fehler mit ungewöhnlichen Mustern (z. B. viele fehlgeschlagene Auth-Versuche)? - Sind neue Credentials angelegt worden?

Bei Abweichung: automatische Benachrichtigung per E-Mail oder Slack.

Option B — Externe Tools:

- **OWASP ZAP:** Kann Webhook-Endpunkte auf bekannte Schwachstellen scannen (öffentlich erreichbare n8n-Endpunkte). Für technisch versierte Teams ohne großen Konfigurationsaufwand nutzbar.
- **Trivy:** Bei Self-hosted-Deployment via Docker — Container-Image auf bekannte CVEs scannen. Einfach in eine CI/CD-Pipeline integrierbar.

Realistische Einschätzung für KMU: Ein einfacher Alert-Workflow in n8n selbst plus ein quartalsweiser manueller Review nach dieser Checkliste ist für die meisten Unternehmen ohne dedizierten Security-Analysten der pragmatischste und nachhaltigste Ansatz.

Beispiel-Workflow – Täglicher Credential-Audit per n8n

Knotenstruktur des Monitoring-Workflows:

1. **Cron-Node** — täglich um 07:00 Uhr ausführen
2. **HTTP-Request-Node** — `GET /credentials` via n8n API (API-Key mit Read-Only-Berechtigung)
3. **Code-Node** — aktuellen Credential-Stand mit gespeicherter Baseline vergleichen; neue oder gelöschte Einträge markieren
4. **IF-Node** — Änderungen vorhanden?
5. **E-Mail- oder Slack-Node** — Alert mit Liste der Änderungen an den zuständigen Admin

---

DSGVO-Checkliste für n8n-Workflows mit personenbezogenen Daten

Kopierfertig für Ihr internes Sicherheitsaudit:

- [ ] Verarbeitungsverzeichnis (Art. 30 DSGVO) enthält alle n8n-Workflows, die personenbezogene Daten verarbeiten
- [ ] Auftragsverarbeitungsvertrag (AVV) mit dem Hosting-Anbieter vorhanden und aktuell
- [ ] Datenminimierung geprüft: Werden nur die tatsächlich notwendigen Felder weitergeleitet?
- [ ] Execution-Logs: Werden Logs mit personenbezogenen Daten automatisch nach einer definierten Frist bereinigt?
- [ ] Zugriffskontrolle: Können nur autorisierte Nutzer Workflows einsehen und bearbeiten?
- [ ] TLS: Sind alle Webhook-Endpunkte ausschließlich über HTTPS erreichbar?
- [ ] Webhook-Authentifizierung: Haben alle produktiven Webhooks einen Authentifizierungsmechanismus?
- [ ] Credentials: Kein API-Key oder Passwort hart im Workflow-Code kodiert?
- [ ] Rotationsplan für Credentials vorhanden und dokumentiert?
- [ ] Incident-Response-Plan: Gibt es einen definierten Ablauf für den Fall eines kompromittierten Workflows?
- [ ] NIS2-Relevanz geprüft: Fällt Ihr Unternehmen unter die Anforderungen des NISG 2026?
- [ ] Inaktive Workflows deaktiviert oder gelöscht?

Eine weiterführende Orientierung zum DSGVO-konformen KI-Einsatz insgesamt bietet der Artikel KI DSGVO-konform einsetzen: Leitfaden für KMU.

Praxis-Tipp – Wann Sie externe Unterstützung brauchen

Viele der beschriebenen Maßnahmen können KMU intern umsetzen: Webhook-Authentifizierung aktivieren, Credential-Review durchführen, einen einfachen Monitoring-Workflow aufsetzen. Das erfordert IT-Grundverständnis, aber kein Sicherheits-Spezialwissen.

Es gibt aber Situationen, in denen externe Fachkompetenz sinnvoll oder notwendig ist:

- Ihre Workflows verarbeiten **Gesundheitsdaten, Zahlungsinformationen** oder haben direkten Schreibzugriff auf Produktivsysteme
- Sie haben **keine dokumentierte Sicherheitsstrategie** und wissen nicht, wo Sie anfangen sollen
- Eine **NIS2-Prüfung** oder ein **Kunden-Audit** steht bevor und n8n ist Teil der geprüften Infrastruktur
- Es gab bereits einen **Sicherheitsvorfall** oder einen Verdacht auf unbefugten Zugriff
- Sie betreiben n8n für **externe Kunden** und tragen damit indirekt deren Datenschutzverantwortung

Ein erstes Gespräch reicht in vielen Fällen aus, um die kritischsten Punkte zu identifizieren und einen realistischen nächsten Schritt zu definieren — ohne daraus sofort ein großes Projekt machen zu müssen.

FAQ

Ist n8n Cloud sicherer als eine Self-hosted Installation?

Nicht pauschal. Cloud entlastet bei Infrastruktur-Sicherheit (Patching, Netzwerk, Verfügbarkeit), aber Workflow-Logik, Credentials und Datenpfade bleiben vollständig Ihre Verantwortung. Self-hosted gibt mehr Kontrolle, erfordert aber aktives Sicherheitsmanagement. Welche Option besser passt, hängt von Ihren IT-Ressourcen und der Datensensibilität ab.

Muss ich für jeden n8n-Workflow einen Penetrationstest beauftragen?

Nein. Ein professioneller Pentest ist bei hochsensiblen Anwendungen sinnvoll — etwa bei Workflows mit Zugriff auf Gesundheitsdaten oder Zahlungssysteme. Für die meisten KMU-Workflows reicht ein strukturierter interner Review nach dieser Checkliste als Basis.

Wie oft sollte ich meine n8n-Sicherheit prüfen?

Mindestens quartalsweise. Zusätzlich: nach jeder größeren Workflow-Änderung, nach einem n8n-Update und bei Personalwechsel mit n8n-Zugang. Wichtige Credential-Rotationen sollten nicht vom Review-Zyklus abhängen, sondern fest im Kalender stehen.

Was passiert bei einem Datenleck durch einen n8n-Workflow?

Es gelten die DSGVO-Meldepflichten: 72 Stunden Meldefrist an die zuständige Datenschutzbehörde (in Österreich die DSB, in Deutschland die jeweilige Landesbehörde). Die Verantwortung liegt beim Unternehmen als Verantwortlichem im Sinne der DSGVO — nicht bei n8n als Tool-Anbieter. Nach aktueller Rechtslage gilt das unabhängig davon, ob Sie Self-hosted oder Cloud nutzen.

Kann ich n8n selbst absichern oder brauche ich externe Hilfe?

Viele Maßnahmen — Webhook-Auth, Credential-Review, Execution-Log-Bereinigung, einfacher Monitoring-Workflow — können intern umgesetzt werden. Bei komplexen Infrastrukturen, regulierten Branchen (z. B. Gesundheit, Finanzdienstleistung) oder fehlendem IT-Knowhow ist externe Unterstützung empfehlenswert.

Nächste Schritte

Die fünf Schritte in der Zusammenfassung: Bestandsaufnahme → Webhook-Tests → SSRF-Prüfung → Credential-Audit → Automatisiertes Monitoring. Das ist der vollständige Prüfprozess für eine n8n-Instanz, die produktiv läuft.

Beginnen Sie mit Schritt 1. Die Bestandsaufnahme kostet keinen Einkauf, kein Tool-Budget und keine externe Hilfe — und sie zeigt in den meisten Fällen sofort, wo der dringendste Handlungsbedarf liegt. Danach können Sie entscheiden, was Sie intern umsetzen und wo Sie Unterstützung brauchen.

Sicherheit in n8n ist kein einmaliges Projekt. Workflows ändern sich, Credentials altern, neue Teammitglieder erhalten Zugang — ein quartalsweiser Review ist deshalb keine Bürokratie, sondern eine realistische Grundlage für kontrollierten Betrieb.

Wenn Sie unsicher sind, ob Ihre n8n-Umgebung sicher genug aufgestellt ist, oder wenn Sie einfach einen zweiten Blick auf Ihre konkrete Situation wollen: Im kostenlosen Beratungsgespräch von Strukturaflow schauen wir gemeinsam auf Ihre Workflows, Ihre Credential-Struktur und Ihre DSGVO-Pflichten — und Sie gehen mit einem klaren nächsten Schritt raus, nicht mit einem Angebot für ein Großprojekt.