Trabalho Final - SAST para detecção de vulnerabilidades em workflows do n8n

João Pedro Schmidt Cordeiro

Tópicos Especiais em Aplicações Tecnológicas I (UFSC-INE5448)

14 de outubro de 2025

Sumário

jctechconnections@gmail.com src/

1 Introdução

A crescente adoção de plataformas Low-Code/No-Code (LCNC) representa uma transformação fundamental no desenvolvimento de aplicações e automações empresariais (N8N.IO, 2025b). A n8n, uma proeminente plataforma de automação de workflows de código aberto, exemplifica essa mudança ao capacitar equipes técnicas a conectar APIs, bancos de dados e serviços através de um editor visual intuitivo (N8N.IO, 2025f). Este trabalho propõe o desenvolvimento de uma ferramenta de Análise Estática de Segurança de Aplicações (SAST) específica para workflows da plataforma n8n, focando na detecção automatizada de vulnerabilidades de segurança em configurações JSON de workflows.

O interesse por este tema surge de uma necessidade prática identificada no ambiente de trabalho, onde utilizo a plataforma n8n diariamente. Com o crescimento do número de workflows e a expansão do uso da ferramenta para diferentes áreas organizacionais, observei a necessidade crítica de averiguar a segurança dos workflows desenvolvidos. Particularmente preocupante é o fato de que muitos usuários que criam workflows não possuem conhecimento técnico aprofundado em segurança, o que pode resultar no desenvolvimento não intencional de vulnerabilidades dentro do sistema. Esta experiência prática evidencia a lacuna de governança identificada na literatura, onde o "desenvolvedor cidadão" assume responsabilidades de desenvolvimento sem o treinamento formal necessário para identificar riscos de segurança.

A experiência prévia com as tecnologias envolvidas no projeto fundamenta-se no uso cotidiano da plataforma n8n para automação de processos empresariais, proporcionando compreensão prática da estrutura JSON de workflows e dos padrões de configuração mais comuns (N8N.IO, 2025a). O conhecimento em análise de código e ferramentas de segurança, combinado com experiência em desenvolvimento de scripts para análise de dados estruturados, oferece a base técnica necessária para implementar soluções de SAST. Além disso, a familiaridade com conceitos de DevSecOps e integração de ferramentas de segurança em pipelines de desenvolvimento, adquirida através de projetos acadêmicos e profissionais, complementa o conjunto de habilidades requerido para o desenvolvimento da ferramenta proposta.

A viabilidade de implementação do MVP é assegurada pela estratégia de escopo limitado e pela reutilização inteligente das ferramentas de estado da arte identificadas na pesquisa. Conforme demonstrado na revisão da literatura, tanto o Agentic Radar quanto o Semgrep possuem arquiteturas comprovadamente funcionais para análise de workflows n8n (SPLXAI, 2025). O Agentic Radar já demonstrou a viabilidade técnica de analisar o JSON de workflows da n8n, construir gráficos de fluxo de dados e gerar relatórios de segurança (SPLXAI, 2025), enquanto o Semgrep oferece um motor de análise estática poderoso e flexível para detecção de padrões customizados (SEMGREP, 2025b). A estratégia proposta consiste em utilizar essas duas ferramentas em conjunto: o Agentic Radar cobrirá vulnerabilidades específicas de IA agêntica, enquanto regras customizadas no Semgrep expandirão a cobertura para vulnerabilidades tradicionais de aplicações web (SQL Injection, SSRF, etc.) que estão além do escopo atual do Agentic Radar. Esta abordagem híbrida permite focar no desenvolvimento de regras específicas e integração, ao invés de construir um motor de análise do zero, tornando o projeto viável dentro do prazo intensivo de 6 semanas estabelecido. O cronograma comprimido exige uma abordagem ágil e focada, priorizando funcionalidades essenciais e mantendo o escopo bem definido.

O potencial de impacto no contexto brasileiro é particularmente significativo considerando a crescente digitalização de processos empresariais e a adoção de ferramentas de automação no país. O Brasil, como um dos maiores mercados de tecnologia da América Latina, tem experimentado um crescimento substancial na adoção de plataformas LCNC, especialmente em setores como serviços financeiros, e-commerce e governo digital. A Lei Geral de Proteção de Dados (LGPD) e outras regulamentações de segurança cibernética no país criam uma demanda específica por ferramentas que possam garantir a conformidade e segurança de automações empresariais. Uma ferramenta SAST especializada para n8n pode contribuir significativamente para elevar o nível de segurança das automações desenvolvidas por organizações brasileiras, reduzindo riscos de vazamento de dados e ataques cibernéticos que podem resultar em penalidades regulatórias e danos reputacionais.

A relevância para a formação profissional está diretamente alinhada com as tendências emergentes do mercado de tecnologia. Conforme evidenciado na revisão da literatura, as plataformas Low-Code/No-Code estão experimentando um crescimento exponencial, com a n8n sendo uma das principais representantes deste paradigma (N8N.IO, 2025f). Este crescimento resulta em uma expansão significativa da superfície de ataque, especialmente considerando que muitos usuários que criam workflows carecem de conhecimento técnico e formação em segurança para prevenir vulnerabilidades básicas e conhecidas (COMMUNITY, 2025b). O desenvolvimento de competências em análise de segurança para plataformas LCNC representa uma especialização altamente demandada no mercado, posicionando o profissional na interseção entre desenvolvimento de baixo código e segurança cibernética. Além disso, o projeto proporciona experiência prática em tecnologias de ponta como SAST, análise de contaminação (taint analysis) e integração de ferramentas de segurança em pipelines de CI/CD, competências essenciais para cargos em DevSecOps e engenharia de segurança. A capacidade de identificar e mitigar riscos em ambientes de desenvolvimento democratizado torna-se um diferencial competitivo crucial à medida que mais organizações adotam estratégias de desenvolvimento cidadão.

A mudança de um modelo de codificação imperativa tradicional para um modelo de configuração declarativa, onde a lógica é definida visualmente e armazenada como objetos JSON, introduz um novo paradigma para a segurança de aplicações. Existe uma lacuna crítica de governança: o "código-fonte" da aplicação (arquivo JSON do workflow) não está sujeito ao mesmo rigor de segurança que a plataforma na qual é executado.

2 Revisão da Literatura e Estado da Arte

Esta seção sintetiza as descobertas da pesquisa acadêmica e análise de soluções existentes na área de SAST para plataformas LCNC.

2.1 A Mudança de Paradigma de Segurança: Do Código Imperativo aos Workflows Declarativos

A ascensão das plataformas de desenvolvimento Low-Code/No-Code (LCNC), como a n8n, representa uma transformação fundamental na forma como as aplicações e automações são construídas (OWASP, 2025e). A n8n, uma proeminente plataforma de automação de workflows de código aberto, capacita equipes técnicas a conectar APIs, bancos de dados e serviços através de um editor visual intuitivo, baseado em nós (N8N.IO, 2025f). Essa abordagem acelera drasticamente o desenvolvimento, permitindo a criação de automações complexas que, de outra forma, exigiriam um esforço de programação significativo (GEEKY-GADGETS, 2025). No entanto, essa mudança de um modelo de codificação imperativa tradicional (como em Python ou Java) para um modelo de configuração declarativa, onde a lógica é definida visualmente e armazenada como objetos JSON, introduz um novo paradigma para a segurança de aplicações.

Neste novo modelo, o perímetro de segurança desloca-se da aplicação principal da plataforma para as configurações criadas pelo usuário. A própria n8n GmbH implementa práticas de segurança robustas em seu código-fonte, incluindo a utilização de Testes de Segurança de Aplicações Estáticas (SAST) como parte de seu pipeline de Integração Contínua/Entrega Contínua (CI/CD) (N8N.IO, 2025j). Essas medidas são projetadas para proteger o motor da n8n contra vulnerabilidades (N8N.IO, 2025i). Contudo, essa varredura não se estende, nem poderia se estender, aos workflows criados pelos seus usuários. A responsabilidade pela concepção de workflows seguros é explicitamente delegada ao usuário, conforme detalhado na documentação oficial (N8N.IO, 2025h).

Isso cria uma lacuna de governança crítica. O "código-fonte"da aplicação, que neste contexto é o arquivo JSON que define o workflow, não está sujeito ao mesmo rigor de segurança que a plataforma na qual ele é executado. O usuário, muitas vezes um "desenvolvedor cidadão"ou um profissional de TI focado na automação de processos, assume o papel de desenvolvedor de aplicações, mas pode não possuir o treinamento formal em segurança necessário para identificar e mitigar riscos complexos (MICHEL, 2025). A consequência é que uma plataforma inerentemente segura ainda pode ser usada para construir e implantar automações perigosamente inseguras. A existência de vulnerabilidades publicadas, como Negação de Serviço (DoS) através de requisições malformadas (FEEDLY, 2025) ou o potencial para Falsificação de Solicitação do Lado do Servidor (SSRF) (PORTSWIGGER, 2025), não reside em falhas no motor principal da n8n, mas na maneira como os workflows podem ser construídos e explorados.

Portanto, o principal desafio de segurança no n8n não é uma falha da plataforma em si, mas sim uma falha potencial na implementação do "usuário-desenvolvedor". As equipes de segurança não podem mais depender exclusivamente das garantias de segurança do fornecedor, como relatórios SOC 2 mencionados em sua documentação legal (N8N.IO, 2025g). Em vez disso, elas devem estabelecer seus próprios processos de garantia para os ativos criados na plataforma. O foco da análise de segurança deve mudar do fornecedor da plataforma para o usuário da plataforma, tratando o workflow JSON como um artefato de código de primeira classe que requer seu próprio processo SAST dedicado para fechar essa lacuna de governança. Esta é uma redefinição fundamental do modelo de responsabilidade compartilhada para plataformas LCNC.

2.2 A Anatomia de um Workflow n8n: Uma Análise da Superfície de Ataque

Para aplicar os princípios de SAST aos workflows da n8n, é imperativo primeiro dissecar sua estrutura fundamental. Cada workflow, independentemente de sua complexidade, pode ser exportado como um único arquivo JSON. Este arquivo não é meramente uma configuração; ele é a representação estática e completa da lógica da aplicação, contendo todos os nós, seus parâmetros e as conexões de fluxo de dados. Para fins de análise de segurança, este JSON é o código-fonte. Uma ferramenta SAST projetada para n8n deve ser capaz de analisar este artefato para identificar padrões inseguros.

A estrutura do JSON de um workflow é composta por vários componentes chave que são cruciais para a análise estática:

- **Array nodes:** Define cada etapa do processamento individual. Uma ferramenta SAST deve iterar sobre este array e analisar cada objeto de nó.
- **Array connections:** Define o gráfico de fluxo de dados do workflow, especificando qual saída de nó se torna a entrada de outro.
- **Expressões dinâmicas:** Como `{{ $json.body.userInput }}`, representam entradas dinâmicas de dados, muitas vezes origem de vulnerabilidades.

A natureza declarativa do JSON da n8n, embora abstraia a complexidade do código tradicional, paradoxalmente torna certos tipos de análise estática mais fáceis e precisos. As ferramentas SAST tradicionais lutam para construir gráficos de fluxo de controle precisos, o que frequentemente resulta em altos índices de falsos positivos. Em contraste,

o modelo da n8n declara explicitamente as relações de fluxo de dados, permitindo regras de análise de contaminação com elevado grau de confiança e baixa taxa de falsos positivos.

2.3 O Estado da Arte do SAST para n8n: Ferramentas e Metodologias

O cenário de ferramentas para análise de segurança estática de workflows n8n é emergente, mas já apresenta abordagens distintas que podem ser categorizadas em ferramentas dedicadas, adaptáveis e soluções baseadas na própria plataforma.

2.3.1 Ferramentas Dedicadas: Agentic Radar

Atualmente, a ferramenta de código aberto mais proeminente e especificamente projetada para escanear workflows n8n é a **Agentic Radar** (SPLXAI, 2025). Ela analisa o JSON do workflow, identifica nós, mapeia conexões e correlaciona padrões com riscos conhecidos, gerando relatórios visuais e detalhados (SPLXAI, 2025). No entanto, seu foco atual está em riscos de IA agêntica, não cobrindo nativamente vulnerabilidades web tradicionais como SQL Injection ou SSRF.

2.3.2 Ferramentas Adaptáveis: Semgrep e Conjuntos de Regras Personalizados

Uma abordagem flexível envolve o uso de motores SAST genéricos, como o **Semgrep** (SEMGREP, 2025b), que suporta análise de JSON e YAML e permite regras personalizadas (SEMGREP, 2025a). Equipes de segurança podem desenvolver conjuntos de regras específicos para o esquema da n8n — por exemplo, detectar concatenações de strings em consultas SQL dentro de nós de banco de dados.

2.3.3 Soluções na Plataforma e Comunitárias

A própria n8n pode ser utilizada para construir ferramentas de segurança, um conceito de "segurança como workflow". Templates como o **WebSecScan** demonstram auditorias automatizadas (N8N.IO, 2025k). A comunidade também tem contribuído com análises manuais, identificando anti-padrões comuns (por exemplo, webhooks públicos sem autenticação) (REDDIT, 2025b).

Esses três grupos de soluções — Agentic Radar, Semgrep e esforços comunitários — representam um ecossistema nascente, mas fragmentado. Há uma oportunidade clara para uma ferramenta que combine o motor de análise consciente da estrutura da n8n com um conjunto de regras abrangente para vulnerabilidades tradicionais.

2.4 Análise Aprofundada de Vulnerabilidades para Workflows n8n

Esta seção detalha as seis principais classes de vulnerabilidades e suas manifestações no contexto da n8n.

2.4.1 SQL Injection (SQLi)

Ocorre quando entradas controladas pelo usuário são inseridas em consultas SQL sem sanitização adequada. No n8n, isso se dá quando dados de gatilhos são passados diretamente para o parâmetro `Query` de nós de banco de dados (COMMUNITY, 2025a). A detecção estática envolve identificar concatenações de expressões `{{...}}` dentro de queries e rastrear sua origem até nós de entrada.

2.4.2 Injeção de Código / Comando

Surge quando dados não confiáveis são usados em comandos executados no servidor. Os nós **Execute Command** e **Code** são os principais vetores (N8N.IO, 2025c). A ferramenta deve rastrear o fluxo de dados para parâmetros como `command` e sinalizar construções dinâmicas baseadas em entradas externas.

2.4.3 Proliferação e Encadeamento de Segredos (Secret Sprawl & Chaining)

Consiste na dispersão de segredos (chaves de API, tokens) em configurações ou logs (GITGUARDIAN, 2025). A análise estática deve procurar padrões de segredos em parâmetros e variáveis, bem como rastrear possíveis vazamentos em respostas de webhooks.

2.4.4 Falsificação de Solicitação do Lado do Servidor (SSRF)

Ocorre quando entradas controladas pelo usuário determinam URLs em nós HTTP Request (N8N.IO, 2025e). A ferramenta deve detectar construções dinâmicas de URLs originadas em fontes externas e verificar se passam por etapas de validação (PORTSWIGGER, 2025).

2.4.5 Manuseio Inseguro de Dados

Abrange falhas na proteção de dados sensíveis durante armazenamento ou transmissão (N8N.IO, 2025l). A ferramenta deve sinalizar uso de HTTP sem TLS, armazenamento não criptografado e fluxos de dados sensíveis para destinos inseguros.

2.4.6 Negação de Serviço (DoS)

Relaciona-se a workflows suscetíveis a loops infinitos ou operações intensivas. A detecção envolve identificar estruturas cíclicas no grafo de conexões e uso de nós vulneráveis a CVEs conhecidos (NIST, 2025).

2.5 Um Contexto Mais Amplo: Mapeando Vulnerabilidades da n8n para o OWASP LCNC Top 10

As vulnerabilidades da n8n refletem riscos sistêmicos descritos pelo **OWASP LCNC Top 10** (OWASP, 2025e). O mapeamento associa, por exemplo, SQL Injection e Command Injection à categoria LCNC-SEC-06 (Injection Handling Failures) (OWASP, 2025b), Secret Sprawl e Data Handling a LCNC-SEC-08 (Data and Secret Handling Failures) (OWASP, 2025d), SSRF a LCNC-SEC-05 (Security Misconfiguration) (OWASP, 2025a) e DoS a LCNC-SEC-07 (Vulnerable and Untrusted Components) (OWASP, 2025c).

Este alinhamento fornece uma linguagem comum para comunicar riscos e priorizar esforços de mitigação em programas corporativos de segurança.

2.6 Uma Estrutura para um Ciclo de Vida de Desenvolvimento de Workflow Seguro (SWDL)

A segurança no n8n deve evoluir da detecção para a prevenção, adotando um ciclo de vida de desenvolvimento seguro de workflows.

2.6.1 Guarda-corpos Automatizados (Shift Left)

Integrar a análise SAST de workflows em pipelines CI/CD é essencial (GITLAB, 2025). Os workflows devem ser escaneados automaticamente a cada commit ou versão.

2.6.2 Padrões de Design e Desenvolvimento Seguro (Um Checklist para Criadores)

Entre as melhores práticas: validação de entradas, não codificar credenciais, princípio do menor privilégio, tratamento robusto de erros, uso de HTTPS e cautela com componentes da comunidade (REDDIT, 2025b).

2.6.3 Auditoria e Monitoramento Contínuo

A segurança requer monitoramento contínuo, auditorias periódicas e análise de logs (OWASP, 2025f). Deve-se estabelecer metas internas (como 0 webhooks públicos) e complementar a análise estática com observabilidade em tempo de execução.

3 Escopo do Projeto

Objetivo Principal: Desenvolver uma ferramenta de Análise Estática de Segurança de Aplicações (SAST) específica para workflows da plataforma n8n, capaz de analisar arquivos JSON e identificar automaticamente vulnerabilidades de segurança. O MVP visa preencher a lacuna de governança identificada, onde o "código-fonte" da aplicação (workflow JSON) não é analisado com o mesmo rigor da plataforma subjacente.

A ferramenta combinará o motor de análise consciente da estrutura da n8n com um conjunto de regras abrangente para vulnerabilidades de aplicações web tradicionais, potencialmente construída sobre o Semgrep (SEMGREP, 2025b), criando uma solução SAST abrangente para o ecossistema.

Componentes do Sistema:

1. **Módulo de Análise de JSON:** Parsing e análise estrutural de arquivos n8n, identificando nós, conexões e expressões dinâmicas (N8N.IO, 2025a).
2. **Engine de Regras de Segurança:** Implementação de regras específicas para as seis classes de vulnerabilidades: SQL Injection, Command Injection, Secret Sprawl, SSRF, Insecure Data Handling e DoS (SEMGREP, 2025a).
3. **Módulo de Análise de Contaminação (Taint Analysis):** Rastreia dados não confiáveis de fontes (webhooks) a sinks perigosos (banco de dados, execução de comandos) (DEV.TO, 2025).
4. **Interface de Relatórios:** Geração de relatórios de segurança e visualização de grafo de fluxo de dados, similar ao Agentic Radar (SPLXAI, 2025).

Limites do Projeto:
Incluído no MVP:

- Análise estática de arquivos JSON.
- Detecção das seis classes principais de vulnerabilidades.
- Relatórios básicos em formato texto/JSON.
- Conjunto inicial de regras personalizadas.

Excluído do MVP:

- Interface gráfica completa.
- Análise dinâmica ou execução real de workflows.
- Análise de nós da comunidade (CYBERSECURITYUP, 2025).
- Correção automática de vulnerabilidades.
- Otimização de performance de workflows.

Requisitos Técnicos:

- Linguagem: Python.
- Parser JSON robusto.
- Engine de regras via Semgrep (SEMGREP, 2025b).
- Biblioteca de grafos para análise de fluxo de dados.

4 Métricas de Sucesso

4.1 Métricas Quantitativas

- **Precisão da Detecção:** 85% de precisão e 90% de recall, F1-score de 87%.
- **Cobertura de Vulnerabilidades:** 100% das seis classes identificadas.
- **Performance de Análise:** máximo de 30 segundos para workflows até 50 nós, e análise de 100 workflows/hora.
- **Taxa de Falsos Positivos:** até 15%.

4.2 Métricas Qualitativas

Usabilidade:

- CLI intuitiva e bem documentada.
- Execução em até três comandos.

Clareza dos Relatórios:

- Severidade por nível (Crítica, Alta, Média, Baixa).
- Explicações técnicas e sugestões de mitigação.
- Mapeamento OWASP LCNC Top 10.

Integração:

- Compatibilidade com JSON.
- Integração simplificada com pipelines CI/CD.

4.3 Benchmarks Comparativos

Com Ferramentas Genéricas (ex.: Semgrep):

- Comparar taxa de detecção, falsos positivos e tempo de configuração.

Com Análise Manual:

- Comparar resultados com análise de 2.000+ workflows reais (REDDIT, 2025a).

Com Agentic Radar:

- Avaliar cobertura de vulnerabilidades e qualidade de relatórios (SPLXAI, 2025).

Critérios SMART:

- Específicos, mensuráveis, alcançáveis, relevantes e temporais.

5 Cronograma Preliminar

O desenvolvimento será executado ao longo de 6 semanas com foco em atividades essenciais.

5.1 Semanas 1-2: Análise do Agentic Radar e Semgrep

Atividades:

- Análise do código-fonte do Agentic Radar (SPLXAI, 2025).
- Estudo do Semgrep e sua integração (SEMGREP, 2025b).

Entregável: Relatório de análise das ferramentas.

5.2 Semana 3: Coleta de Workflows

Atividades:

- Coleta de 3 workflows reais para análise (REDDIT, 2025a).

Entregável: Dataset com 3 workflows documentados.

5.3 Semanas 4-5: Desenvolvimento

Atividades:

- Protótipo de parser JSON (N8N.IO, 2025d).
- Conjunto de regras de segurança (SEMGREP, 2025a).

Entregáveis:

- Parser JSON funcional.
- Regras de detecção implementadas.

5.4 Semana 6: Conclusões

Atividades:

- Testes com os 3 workflows coletados.
- Conclusões finais e documentação.

Entregável: Relatório final com análise dos resultados.

Referências

COMMUNITY, N. *Massive security vulnerabilities through improper mysql escaping*. 2025. Disponível em: <https://community.n8n.io/t/massive-security-vulnerabilities-through-improper-mysql-escaping/167711>. Acesso em: 23 set. 2025.

COMMUNITY, N. *N8n cloud security: Protection against ddos, unauthorized access & webhook abuse*. 2025. Disponível em: <https://community.n8n.io/t/n8n-cloud-security-protection-against-ddos-unauthorized-access-webhook-abuse/181039>. Acesso em: 23 set. 2025.

CYBERSECURITYUP. *n8n-cybersecurity-workflows: Security automation with n8n ideas: 100+ red/blue/appsec workflows, integrations, and ready-to-run playbooks*. 2025. Disponível em: <https://github.com/CyberSecurityUP/n8n-CyberSecurity-Workflows>. Acesso em: 23 set. 2025.

DEV.TO. *Tryhackme: Prototype pollution - dev community*. 2025. Disponível em: <https://dev.to/seanleeys/tryhackme-prototype-pollution-5fa2>. Acesso em: 23 set. 2025.

FEEDLY. *CVE-2023-27562 - exploits & severity - feedly*. 2025. Disponível em: <https://feedly.com/cve/CVE-2023-27562>. Acesso em: 23 set. 2025.

GEEKY-GADGETS. *30 essential n8n tricks that will take your ai automations to the next level*. 2025. Disponível em: <https://www.geeky-gadgets.com/n8n-features-for-automation/>. Acesso em: 23 set. 2025.

GITGUARDIAN. *Gitguardian introduces one-click secret revocation to accelerate incident response*. 2025. Disponível em: <https://blog.gitguardian.com/gitguardian-introduces-one-click-secret-revocation-to-accelerate-incident-respons>. Acesso em: 23 set. 2025.

GITLAB. *Static application security testing (sast) - gitlab docs*. 2025. Disponível em: <https://docs.gitlab.com/user/application_security/sast/>. Acesso em: 23 set. 2025.

MICHEL, M. *n8n security best practices: Protect your data and workflows*. 2025. Disponível em: <https://mathias.rocks/blog/2025-01-20-n8n-security-best-practices>. Acesso em: 23 set. 2025.

N8N.IO. *AI coding - n8n docs*. 2025. Disponível em: <https://docs.n8n.io/code/ai-code/>. Acesso em: 23 set. 2025.

N8N.IO. *AI workflow automation platform & tools - n8n*. 2025. Disponível em: <https://n8n.io/>. Acesso em: 23 set. 2025.

N8N.IO. *Code*. 2025. Disponível em: <https://docs.n8n.io/integrations/builtin/core-nodes/n8n-nodes-base.code/>. Acesso em: 25 set. 2025.

N8N.IO. *Code integrations | workflow automation with n8n*. 2025. Disponível em: <https://n8n.io/integrations/code/>. Acesso em: 23 set. 2025.

N8N.IO. *HTTP Request*. 2025. Disponível em: <https://docs.n8n.io/integrations/builtin/core-nodes/n8n-nodes-base.httprequest/>. Acesso em: 25 set. 2025.

N8N.IO. *n8n - Workflow automation for technical people*. 2025. Disponível em: <https://n8n.io/>. Acesso em: 25 set. 2025.

N8N.IO. *n8n legal*. 2025. Disponível em: <https://n8n.io/legal/>. Acesso em: 23 set. 2025.

N8N.IO. *Privacy - n8n docs*. 2025. Disponível em: <https://docs.n8n.io/privacy-security/privacy/>. Acesso em: 23 set. 2025.

N8N.IO. *Security - n8n*. 2025. Disponível em: <https://n8n.io/legal/security/>. Acesso em: 23 set. 2025.

N8N.IO. *Security overview · n8n-io/n8n · github*. 2025. Disponível em: <https://github.com/n8n-io/n8n/security>. Acesso em: 23 set. 2025.

N8N.IO. *Websecscan: Ai-powered website security auditor | n8n workflow template*. 2025. Disponível em: <https://n8n.io/workflows/3314-websecscan-ai-powered-website-security-auditor/>. Acesso em: 23 set. 2025.

N8N.IO. *What you can do - n8n docs*. 2025. Disponível em: <https://docs.n8n.io/privacy-security/what-you-can-do/>. Acesso em: 23 set. 2025.

NIST. *CVE-2025-49595 detail - nvd*. 2025. Disponível em: <https://nvd.nist.gov/vuln/detail/CVE-2025-49595>. Acesso em: 23 set. 2025.

OWASP. *LCNC-SEC-05: Security misconfiguration - owasp foundation*. 2025. Disponível em: <https://owasp.org/www-project-top-10-low-code-no-code-security-risks/content/2022/en/LCNC-SEC-05-Security-Misconfiguration>. Acesso em: 23 set. 2025.

OWASP. *LCNC-SEC-06: Injection handling failures | owasp foundation*. 2025. Disponível em: <https://owasp.org/www-project-top-10-low-code-no-code-security-risks/content/2022/en/LCNC-SEC-06-Injection-Handling-Failures>. Acesso em: 23 set. 2025.

OWASP. *LCNC-SEC-07: Vulnerable and untrusted components - owasp foundation*. 2025. Disponível em: <https://owasp.org/www-project-top-10-low-code-no-code-security-risks/content/2022/en/LCNC-SEC-07-Vulnerable-and-Untrusted-Components>. Acesso em: 23 set. 2025.

OWASP. *LCNC-SEC-08: Data and secret handling failures | owasp foundation*. 2025. Disponível em: <https://owasp.org/www-project-top-10-low-code-no-code-security-risks/content/2022/en/LCNC-SEC-08-Data-and-Secret-Handling-Failures>. Acesso em: 23 set. 2025.

OWASP. *Owasp low-code/no-code top 10*. 2025. Disponível em: <https://owasp.org/www-project-top-10-low-code-no-code-security-risks/>. Acesso em: 23 set. 2025.

OWASP. *Source code analysis tools - owasp foundation*. 2025. Disponível em: <https://owasp.org/www-community/Source_Code_Analysis_Tools>. Acesso em: 23 set. 2025.

PORTSWIGGER. *What is ssrf (server-side request forgery)? tutorial & examples | web security academy*. 2025. Disponível em: <https://portswigger.net/web-security/ssrf>. Acesso em: 23 set. 2025.

REDDIT. *I analysed 2000+ n8n workflows and this is what i learned - reddit*. 2025. Disponível em: <https://www.reddit.com/r/n8n/comments/1l1f6n8/i_analysed_2000_n8n_workflows_and_this_is_what_i/>. Acesso em: 23 set. 2025.

REDDIT. *Sharing some best practices for reliable and secure n8n automations - reddit*. 2025. Disponível em: <https://www.reddit.com/r/n8n/comments/1m7wq6q/sharing_some_best_practices_for_reliable_and/>. Acesso em: 23 set. 2025.

SEMGREP. *Custom rule examples - semgrep*. 2025. Disponível em: <https://semgrep.dev/docs/writing-rules/rule-ideas>. Acesso em: 23 set. 2025.

SEMGREP. *Semgrep app security platform | ai-assisted sast, sca and secrets detection*. 2025. Disponível em: <https://semgrep.dev/>. Acesso em: 23 set. 2025.

SPLXAI. *Scanning n8n workflows with agentic radar*. 2025. Disponível em: <https://medium.com/@SplxAI/scanning-n8n-workflows-with-agentic-radar-62f8e1a5c705>. Acesso em: 23 set. 2025.

SPLXAI. *Scanning n8n workflows with agentic radar | splxai blog*. 2025. Disponível em: <https://splx.ai/blog/scanning-n8n-workflows-with-agentic-radar>. Acesso em: 23 set. 2025.