SAST para detecção de vulnerabilidades em workflows do n8n

João Pedro Schmidt Cordeiro Gustavo Zambonin

11 de novembro de 2025

Resumo

A crescente adoção de plataformas Low-Code/No-Code (LCNC), como o n8n, tem transformado o desenvolvimento de automações empresariais, democratizando a criação de workflows através de interfaces visuais. Entretanto, essa mudança de paradigma introduz uma lacuna crítica de governança: enquanto a plataforma subjacente é protegida por práticas robustas de segurança, os workflows criados pelos usuários, armazenados como arquivos JSON declarativos, não são submetidos ao mesmo rigor de análise de segurança. Este trabalho propõe o desenvolvimento de uma ferramenta de Análise Estática de Segurança de Aplicações (SAST) específica para workflows da n8n, capaz de detectar automaticamente vulnerabilidades como SQL Injection, Command Injection, SSRF, exposição de credenciais, manuseio inseguro de dados e negação de serviço. A metodologia adotada baseia-se em uma estratégia híbrida que combina duas ferramentas do estado da arte: o Agentic Radar, focado em riscos específicos de agentes de IA, e o Semgrep, configurado com regras customizadas para vulnerabilidades tradicionais de aplicações web. Além disso, propõem-se abordagens inovadoras para a criação de regras para o Semgrep, utilizando inteligência artificial e workflows com problemas conhecidos como base para o desenvolvimento e validação de padrões de detecção. A análise detalhada revelou que o Agentic Radar oferece cobertura efetiva de 16,7% das classes de vulnerabilidades identificadas, enquanto o Semgrep apresenta potencial de cobertura de 66,7% quando adequadamente configurado. Os resultados demonstram a viabilidade técnica da abordagem proposta, validando que a análise estática de estruturas declarativas JSON não apenas é possível, mas também eficaz para identificação de padrões inseguros. A solução contribui para elevar o nível de segurança das automações desenvolvidas em organizações brasileiras, auxiliando na conformidade com regulamentações como a LGPD e reduzindo riscos de vazamento de dados e ataques cibernéticos em ambientes de desenvolvimento democratizado.

Palavras-chave: SAST, n8n, Low-Code/No-Code, Segurança de Aplicações, Análise Estática, Vulnerabilidades, Automação de Workflows, Inteligência Artificial, LGPD.

Sumário

1 Introdução

[Apresentar o contexto do problema, a motivação para o trabalho, o problema específico abordado, delimitação clara do escopo (o que está incluído e excluído), e as contribuições esperadas da PoC.]

1.1 Motivação e Contexto

[Explicar o cenário atual, as limitações dos métodos existentes, e por que o problema é relevante. Incluir dados estatísticos quando disponíveis.]

1.2 Problema Abordado

[Definir especificamente qual problema técnico será resolvido pela PoC, listando os tipos de situações ou casos de uso que serão tratados.]

1.3 Delimitação do Escopo

[Listar explicitamente o que NÃO será abordado no trabalho para evitar expectativas incorretas.]

1.4 Contribuições Esperadas

[Enumerar as principais contribuições técnicas, metodológicas ou científicas que a PoC pretende demonstrar.]

2 Fundamentação Teórica e Estado da Arte

[Revisar a literatura relevante, técnicas existentes, trabalhos relacionados, e identificar lacunas que justifiquem o desenvolvimento da PoC.]

2.1 Técnicas de [Área Específica]

[Descrever as principais técnicas e abordagens já existentes na área do problema, tanto clássicas quanto modernas.]

2.1.1 Abordagens Clássicas

[Apresentar métodos tradicionais, suas características, vantagens e limitações.]

2.1.2 Abordagens Baseadas em ML/DL

[Descrever técnicas modernas de aprendizado de máquina ou profundo aplicadas ao problema, incluindo arquiteturas de redes neurais relevantes.]

2.2 Explicabilidade em Modelos de IA

[Discutir técnicas de XAI (Explainable AI) relevantes para o projeto, como Grad-CAM, SHAP, LIME, e sua importância no contexto da aplicação.]

2.3 Segurança e Privacidade em Sistemas de IA

[Abordar vulnerabilidades comuns em sistemas de IA, ataques adversariais, e requisitos de conformidade legal (LGPD, GDPR, etc.).]

2.3.1 Ataques Adversariais

[Explicar tipos de ataques (white-box, black-box), técnicas de ataque (FGSM, PGD) e possíveis defesas.]

2.3.2 Conformidade com LGPD

[Detalhar os artigos relevantes da LGPD e como eles se aplicam ao projeto específico.]

2.4 Gaps e Oportunidades

[Identificar lacunas na literatura e oportunidades de pesquisa que justifiquem o desenvolvimento da PoC.]

3 Objetivos e Métricas de Sucesso

[Definir claramente os objetivos gerais e específicos do projeto, estabelecendo métricas quantitativas e qualitativas para avaliar o sucesso da PoC.]

3.1 Objetivo Geral

[Declarar o objetivo principal da PoC de forma clara e mensurável, incluindo metas de performance esperadas.]

3.2 Objetivos Específicos

[Listar objetivos técnicos detalhados, numerados e verificáveis, cobrindo aspectos de implementação, segurança, privacidade e conformidade.]

OE1: *[Objetivo específico 1]*

OE2: *[Objetivo específico 2]*

OE3: *[Objetivo específico 3]*

3.3 Métricas de Avaliação

[Estabelecer critérios objetivos para avaliar o sucesso do projeto.]

3.3.1 Métricas Quantitativas

[Definir métricas numéricas (acurácia, F1-score, latência, throughput, etc.) com valores-alvo e justificativas para as metas estabelecidas.]

Tabela 1: Métricas quantitativas e valores-alvo

Métrica	Definição	Meta

3.3.2 Métricas Qualitativas

[Estabelecer critérios subjetivos de avaliação (usabilidade, clareza de explicações, facilidade de integração) e métodos de avaliação.]

3.3.3 Métricas de Robustez e Segurança

[Definir testes específicos de robustez (transformações, compressões, ataques adversariais) com critérios de aceitação.]

Tabela 2: Testes de robustez obrigatórios

Cenário	Objetivo

3.4 Benchmarks Comparativos

[Identificar sistemas ou trabalhos que servirão como baseline para comparação de performance.]

4 Escopo da Solução e Requisitos

[Detalhar os casos de uso, arquitetura do sistema, requisitos técnicos e composição dos dados.]

4.1 Casos de Uso

[Descrever os principais fluxos de interação com o sistema.]

4.1.1 Caso de Uso Principal

[Detalhar o cenário principal de uso incluindo atores, pré-condições, fluxo principal e fluxos alternativos.]

Ator primário: *[Nome do ator]*

Pré-condições:

- *[Pré-condição 1]*
- *[Pré-condição 2]*

Fluxo principal:

1. *[Passo 1]*
2. *[Passo 2]*
3. *[Passo 3]*

Fluxos alternativos:

- *[Alternativa 1]*
- *[Alternativa 2]*

4.1.2 Casos Excluídos do Escopo

[Listar explicitamente casos de uso que NÃO serão implementados na PoC.]

4.2 Arquitetura Proposta

[Apresentar a arquitetura lógica do sistema, preferencialmente com diagramas, explicando as camadas, componentes e fluxo de dados. Incluir considerações de segurança (defense in depth, privacy by design).]

4.2.1 Componentes Principais

[Descrever em detalhes cada componente da arquitetura (frontend, backend, modelo de ML, sistema de logs, etc.) e suas responsabilidades.]

4.3 Requisitos Técnicos

[Especificar as tecnologias, frameworks e infraestrutura necessários.]

4.3.1 Tecnologias e Frameworks

[Listar em formato de tabela as tecnologias escolhidas para cada camada/componente com justificativas técnicas.]

Tabela 3: Stack tecnológico

Camada	Tecnologia	Justificativa

4.3.2 Infraestrutura Mínima

[Especificar requisitos de hardware (CPU, GPU, RAM, disco) e rede para desenvolvimento, treinamento e deploy.]

4.3.3 Dados de Treinamento e Validação

[Descrever a composição do dataset (quantidade, proporções, fontes), estratégia de divisão (treino/validação/teste) e considerações de privacidade.]

Tabela 4: Composição do dataset

Conjunto	**Classe A**	**Classe B**	**Total**
Treinamento (70%)			
Validação (15%)			
Teste (15%)			
Total			

5 Ameaças, Riscos e Controles

[Aplicar metodologia STRIDE para identificação sistemática de ameaças, analisar riscos específicos de sistemas de ML, e estabelecer controles de mitigação.]

5.1 Superfícies de Ataque

[Enumerar todos os pontos do sistema que podem ser explorados por atacantes.]

5.2 Análise STRIDE

[Apresentar matriz completa de ameaças usando categorias STRIDE (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege) com controles correspondentes.]

Tabela 5: Matriz de ameaças STRIDE

Categoria	**Ameaça**	**Controle**
Spoofing		
Tampering		
Repudiation		
Information Disclosure		
Denial of Service		
Elevation of Privilege		

5.3 Ataques Específicos a Modelos de ML

[Detalhar ameaças particulares a sistemas de aprendizado de máquina.]

5.3.1 Evasion Attacks

[Descrever ataques adversariais que tentam enganar o modelo em tempo de inferência e controles de mitigação.]

5.3.2 Data Poisoning

[Explicar riscos de contaminação do conjunto de treinamento e medidas preventivas.]

5.3.3 Model Extraction

[Abordar riscos de roubo de modelo proprietário através de queries e contramedidas.]

5.4 Considerações de Privacidade (LINDDUN)

[Aplicar metodologia LINDDUN para análise sistemática de riscos de privacidade (Linkability, Identifiability, Non-repudiation, Detectability, Disclosure, Unawareness, Non-compliance).]

5.5 Considerações Éticas

[Discutir aspectos éticos do sistema.]

5.5.1 Vieses Algorítmicos

[Identificar riscos de discriminação, estratégias de mitigação e métodos de avaliação de equidade.]

5.5.2 Transparência e Explicabilidade

[Estabelecer compromissos de transparência e como a explicabilidade será implementada.]

5.5.3 Direito de Contestação

[Definir processo para usuários contestarem decisões automatizadas.]

6 Metodologia de Desenvolvimento da PoC

[Detalhar todos os aspectos práticos de implementação: preparação de dados, modelagem, treinamento, implementação e integração.]

6.1 Dados e Preparação

[Descrever origem, geração e preparação dos dados.]

6.1.1 Geração de Dataset Sintético (ou Coleta de Dados Reais)

[Explicar como os dados foram obtidos ou gerados, incluindo scripts, ferramentas e técnicas utilizadas.]

6.1.2 Particionamento e Estratificação

[Detalhar como o dataset foi dividido, garantindo representatividade em todos os subconjuntos.]

6.2 Modelagem e Treinamento

[Explicar escolhas de arquitetura e processo de treinamento.]

6.2.1 Arquiteturas Avaliadas

[Listar modelos candidatos com características técnicas (número de parâmetros, FLOPS, pré-treinamento).]

Tabela 6: Modelos candidatos

Modelo	Parâmetros	FLOPS	Pré-treinamento

6.2.2 Estratégia de Treinamento

[Detalhar hiperparâmetros, fases de treinamento (transfer learning, fine-tuning), técnicas de data augmentation, regularização, função de perda e otimizadores utilizados. Incluir snippets de código relevantes.]

6.2.3 Adversarial Training

[Explicar como exemplos adversariais foram incorporados ao treinamento para aumentar robustez, incluindo código de implementação.]

6.3 Implementação e Integração

[Descrever aspectos práticos de implementação do sistema completo.]

6.3.1 Pipeline de Inferência

[Apresentar código do pipeline completo de processamento, desde entrada até geração de resultados e logs.]

6.3.2 API REST

[Documentar endpoints, formatos de request/response, autenticação e exemplos de uso via curl ou outra ferramenta.]

6.3.3 Containerização

[Apresentar configuração Docker/Docker Compose para reprodutibilidade e deploy.]

7 Avaliação Experimental

[Apresentar protocolo experimental, resultados obtidos, comparações e análises.]

7.1 Protocolo Experimental

[Detalhar configuração de hardware/software e metodologia de avaliação.]

7.1.1 Configuração de Hardware e Software

[Especificar ambiente de execução (GPU, RAM, OS, versões de bibliotecas).]

7.1.2 Repetição e Validação Cruzada

[Explicar como a variância foi tratada (múltiplas execuções, seeds diferentes, intervalos de confiança).]

7.2 Resultados

[Apresentar resultados quantitativos e qualitativos.]

7.2.1 Performance no Conjunto de Teste

[Tabela com métricas principais (F1, Precisão, Recall, AUC-ROC) por modelo, comparando com metas estabelecidas.]

Tabela 7: Métricas principais por modelo

Modelo	F1-Score	Precisão	Recall	AUC-ROC
Meta Alvo				

7.2.2 Matriz de Confusão

[Apresentar matriz de confusão com análise de falsos positivos e falsos negativos, identificando padrões nos erros.]

7.2.3 Performance por Tipo/Categoria

[Detalhar desempenho em subcategorias do problema para identificar pontos fortes e fracos.]

7.2.4 Latência e Throughput

[Reportar métricas de tempo de processamento e vazão, com breakdown por etapa do pipeline.]

7.3 Testes de Robustez

[Avaliar resiliência do sistema.]

7.3.1 Transformações Geométricas e Degradações

[Tabela mostrando degradação de performance sob transformações comuns (compressão, ruído, rotação).]

7.3.2 Ataques Adversariais

[Resultados de testes contra ataques adversariais (FGSM, PGD) com diferentes intensidades.]

7.4 Análise de Vieses

[Avaliar equidade do sistema.]

7.4.1 Equidade entre Subgrupos

[Tabela comparando performance entre diferentes subgrupos para identificar possíveis discriminações.]

7.5 Explicabilidade: Análise Qualitativa

[Avaliar qualidade das explicações geradas (Grad-CAM ou outra técnica) através de avaliação por especialistas ou métricas objetivas. Incluir exemplos visuais.]

8 Discussão

[Interpretar resultados, discutir limitações, comparar com estado-da-arte e analisar implicações práticas.]

8.1 Alcance dos Objetivos

[Avaliar se os objetivos estabelecidos foram atingidos, comparando resultados com metas.]

8.2 Limitações Identificadas

[Discutir honestamente as limitações do trabalho.]

8.2.1 Limitações Técnicas

[Identificar restrições técnicas (dependência de qualidade de entrada, escopo limitado, dataset sintético, etc.).]

8.2.2 Limitações de Segurança

[Reconhecer vulnerabilidades residuais e aspectos de segurança que precisam ser melhorados para produção.]

8.3 Comparação com Estado-da-Arte

[Tabela comparativa com trabalhos relacionados e sistemas comerciais, analisando posicionamento da PoC.]

8.4 Trade-offs Segurança vs. Usabilidade

[Discutir compromissos feitos entre segurança, privacidade e usabilidade, justificando escolhas.]

8.5 Implicações Práticas

[Analisar impactos práticos do trabalho.]

8.5.1 Para Usuários/Stakeholders

[Discutir como o sistema pode ser utilizado na prática, benefícios e necessidades de treinamento.]

8.5.2 Para Políticas Públicas

[Refletir sobre implicações para regulamentação, padronização e investimentos públicos.]

9 Conclusões e Próximos Passos

[Sintetizar contribuições, propor trabalhos futuros e fornecer recomendações.]

9.1 Síntese das Evidências

[Resumir as principais conclusões demonstradas pela PoC de forma objetiva.]

9.2 Contribuições

[Listar contribuições metodológicas, técnicas e científicas do trabalho.]

9.3 Trabalhos Futuros

[Propor extensões e melhorias.]

9.3.1 Curto Prazo (3-6 meses)

[Melhorias incrementais possíveis em 3-6 meses.]

9.3.2 Médio Prazo (6-12 meses)

[Desenvolvimento de funcionalidades adicionais ou pilotos em 6-12 meses.]

9.3.3 Longo Prazo (1-2 anos)

[Visão de evolução do sistema em 1-2 anos (escala, novas funcionalidades, federação).]

9.4 Recomendações

[Fornecer orientações práticas para stakeholders interessados em adotar tecnologia similar.]

10 Checklist de Conformidade (LGPD/Ética/Segurança)

10.1 Lei Geral de Proteção de Dados (LGPD)

- ☐ **Minimização de dados (Art. 6º, III):**
  - ☐ *[Item de verificação 1]*
  - ☐ *[Item de verificação 2]*
- ☐ **Finalidade específica (Art. 6º, I):**
  - ☐ *[Item de verificação 1]*
- ☐ **Transparência (Art. 6º, VI):**
  - ☐ *[Item de verificação 1]*
- ☐ **Segurança (Art. 46):**
  - ☐ *[Item de verificação 1]*
- ☐ **Direito à explicação (Art. 20):**
  - ☐ *[Item de verificação 1]*
- ☐ **Não discriminação (Art. 6º, IX):**
  - ☐ *[Item de verificação 1]*

10.2 Segurança da Informação

- ☐ **Criptografia:**
  - ☐ *[Item de verificação 1]*
- ☐ **Autenticação e Autorização:**
  - ☐ *[Item de verificação 1]*
- ☐ **Isolamento:**
  - ☐ *[Item de verificação 1]*
- ☐ **Auditoria:**
  - ☐ *[Item de verificação 1]*

- ☐ **Resiliência:**
  - ☐ *[Item de verificação 1]*

10.3 Ética e IA Responsável

- ☐ **Explicabilidade:**
  - ☐ *[Item de verificação 1]*
- ☐ **Equidade:**
  - ☐ *[Item de verificação 1]*
- ☐ **Accountability:**
  - ☐ *[Item de verificação 1]*
- ☐ **Contestação:**
  - ☐ *[Item de verificação 1]*
- ☐ **Transparência pública:**
  - ☐ *[Item de verificação 1]*

10.4 Ações Pendentes para Produção

- ☐ *[Ação pendente 1]*
- ☐ *[Ação pendente 2]*

Agradecimentos

[Agradecer aos orientadores, colegas, instituições e fontes de financiamento quando aplicável.]

A Detalhes de Implementação

A.1 Estrutura de Diretórios

[Apresentar árvore completa de diretórios do projeto.]

A.2 Comandos para Reprodução

[Lista step-by-step de comandos para setup, treinamento, avaliação e deploy do sistema.]

B Exemplos Adicionais de Resultados

[Figuras adicionais mostrando casos de sucesso e falha, exemplos visuais de classificações corretas e incorretas.]

C Código-fonte Completo

[Link para repositório público com código-fonte e licença de uso.]