

Framework de IA para Detecção e Mitigação de Ameaças em Redes Móveis Privativas

Gabriel Franco¹, Hiuri Noronha², Gabriel Baltazar¹, Clodoaldo Callogero¹, Marcos Tomita¹, Vanessa Testoni³, Leandro Manera³, Luciano Martins⁴, Rodolfo de Sousa Costa⁴

{hi.gabrielhfranco, clodorc, gabriel.baltazar.br marcos.tomita}@gmail.com¹
 hiuri.noronha@ieee.org²
 {vtestoni, lemanera}@unicamp.br³
 {lmartins, rcosta}@cpqd.br⁴

Resumo—Redes móveis privativas exigem mecanismos de segurança capazes de operar em ambientes distribuídos. Este artigo propõe um framework de IA para detecção e mitigação de ameaças em uma arquitetura com rede de acesso por rádio remota via enlace satelital de baixa órbita. A solução integra as ferramentas Wazuh, Suricata, scripts em Python, a plataforma de automação n8n e modelos de linguagem para correlacionar eventos e apoiar respostas semiautomatizadas, incluindo bloqueio de IPs suspeitos e envio de notificações. Em um cenário controlado de negação de serviço, o sistema correlacionou 28 alertas em 24 horas e aplicou contenção em até 1 minuto, demonstrando viabilidade operacional.

Palavras-Chave—Redes móveis privativas; segurança de redes; inteligência artificial; sistemas multiagentes.

Abstract—Private mobile networks require security mechanisms capable of operating in distributed environments. This paper proposes an AI-based framework for threat detection and mitigation in an architecture with a remote radio access network over a low Earth orbit satellite link. The proposed solution integrates security tools such as Wazuh and Suricata, along with Python scripts, the N8N automation platform, and large language models to correlate events and support semi-automated responses, including blocking suspicious IP addresses and sending notifications. In a controlled denial-of-service scenario, the system correlated 28 alerts within 24 hours and applied containment actions within up to 1 minute, demonstrating the operational feasibility of the proposed approach.

Keywords—Private mobile networks; network security; artificial intelligence; multi-agent systems.

I. INTRODUÇÃO

As redes móveis privativas têm se consolidado como alternativa para prover conectividade dedicada a organizações com requisitos específicos de cobertura, disponibilidade e controle operacional. Na padronização do 5G, o 3GPP define redes não públicas (NPN) como redes destinadas ao uso exclusivo de uma entidade privada, podendo ser implantadas como redes totalmente *standalone*, hospedadas por uma rede móvel terrestre pública (PLMN) ou oferecidas como *slice* de uma PLMN [1]. No contexto brasileiro, a Anatel caracteriza redes privativas como redes de telecomunicações voltadas a soluções

específicas para uso próprio ou de grupos de usuários [2], e trabalhos recentes têm investigado testbeds 5G privativos sob diferentes configurações e métricas de desempenho [3].

Embora o 5G disponha de um arcabouço normativo robusto de segurança [4], redes privativas frequentemente operam com dados sensíveis e em cenários de implantação heterogêneos, o que amplia a necessidade de garantias de confidencialidade, integridade e disponibilidade [5]. Esse desafio é ainda mais relevante em topologias distribuídas, nas quais a presença de uma rede de acesso por rádio remota (RAN) e a dependência de domínios externos de transporte, como enlaces satelitais, podem ampliar a superfície de ataque e comprometer a continuidade do serviço [5].

Na prática, a operação de segurança em redes privativas depende de instrumentos de detecção e monitoramento, como sistemas de detecção de intrusão (IDS) e correlação de eventos de segurança por meio de plataformas de Gerenciamento de Informações e Eventos de Segurança (SIEM), em geral orientadas por regras e assinaturas. Soluções abertas como Wazuh e Suricata são comumente combinadas para enriquecer a detecção, integrando telemetria de *endpoints* e inspeção de tráfego [6], [7], [8]. Entretanto, a literatura aponta desafios persistentes, como escalabilidade, grande volume de dados e alarmes espúrios, ao mesmo tempo em que evidencia o potencial de modelos de linguagem de grande porte (LLMs) para ampliar a interpretabilidade e a eficiência da análise de logs e da triagem de ameaças [9]. Além disso, diretrizes atuais de resposta a incidentes reforçam a necessidade de integrar detecção, análise e contenção em fluxos auditáveis e reproduzíveis [10].

Embora as soluções tradicionais de IDS/SIEM, fundadas em regras e assinaturas estáticas, assegurem visibilidade operacional, apresentam limitações como elevado volume de alarmes espúrios, dificuldade de correlacionar eventos dispersos e dependência de triagem manual. A literatura indica que modelos de linguagem de grande porte ampliam a interpretabilidade dos registros e reduzem o tempo de análise, enquanto a recuperação aumentada de geração (RAG) confere memória de longo prazo, permitindo recuperar incidentes similares e qualificar a recomendação de contenção [9]. Tais capacidades sugerem vantagem qualitativa da abordagem orientada por IA, cuja comparação quantitativa permanece em aberto.

Os autores agradecem o apoio do MCTI via FUNTTEL/FINEP (Projeto ERA 6G, Contrato nº 01.26.0141.00 - Referência 2124/25) e PPI SOFTEX/Lei 8.248/91 (Residência em TIC 11, DOU 01245.011733/2022-83), bem como ao CPQD.

Diante desse contexto, este artigo propõe e valida um framework orientado por IA para detecção e mitigação de ameaças em uma rede móvel privativa com RAN remota e *backhaul* satelital. A proposta integra eventos oriundos de Wazuh e Suricata em um fluxo automatizado de análise, emprega LLMs e recuperação de contexto via RAG para apoiar a interpretação dos incidentes, e estrutura uma resposta com mediação humana e execução de ações de contenção. Para favorecer a replicabilidade, os incidentes são mapeados ao MITRE ATT&CK, e o fluxo de resposta é alinhado a recomendações atuais de resposta a incidentes [10]. As contribuições centrais incluem uma arquitetura que integra de ponta a ponta as ferramentas Wazuh e Suricata, a orquestração no n8n com agentes de linguagem especializados em diagnóstico, comunicação, validação e execução, a memória de longo prazo implementada com RAG, o servidor MCP como camada de mediação, provendo as ferramentas de execução para os agentes, e a contenção no firewall com confirmação humana. O artigo visa ainda demonstrar que o encadeamento de agentes em fluxo único e auditável permite separar diagnóstico técnico, comunicação e resposta operacional, com validação em topologia realista com RAN remota e *backhaul* satelital, em ataque controlado de negação de serviço (DoS).

II. FRAMEWORK PROPOSTO E AMBIENTE DE VALIDAÇÃO

A. Ambiente Experimental

O ambiente de validação está inserido em uma arquitetura de rede móvel privativa com núcleo centralizado e integração com uma RAN remota por meio de um enlace satelital de baixa órbita (LEO), conforme ilustrado na Figura 1. A adoção desse *backhaul* busca representar um cenário realista de conectividade para locais remotos, nos quais a integração da RAN ao núcleo depende de infraestrutura de transporte heterogênea. Constelações de satélites de baixa órbita têm-se configurado como uma alternativa consolidada de *backhaul* para redes móveis em áreas remotas, nas quais a infraestrutura terrestre de transporte é limitada. No domínio central encontram-se o serviço de *core* e os elementos de controle da infraestrutura. No lado remoto, a rede de acesso é composta por uma estação rádio base (ERB) Baicells, equipamentos de usuário (UEs) e um roteador MikroTik, responsável pelo encaminhamento do tráfego local e pela interconexão com o ambiente central. O enlace LEO provê a conectividade entre os domínios, enquanto um firewall pfSense atua como ponto de borda da infraestrutura.

A interligação segura entre a rede remota e o núcleo foi realizada por meio de um túnel OpenVPN entre o MikroTik e o pfSense, permitindo transporte protegido do tráfego e roteamento entre segmentos distintos sem uso de NAT. A topologia também adotou segmentação lógica entre os planos de controle e de usuário, com redes distintas nos domínios central e remoto, além de uma sub-rede dedicada ao túnel VPN.

Para o monitoramento contínuo e a análise de eventos, a infraestrutura *on-premise* conta com um servidor virtualizado em Proxmox. Nesse ambiente, foram instanciadas máquinas virtuais dedicadas a monitoramento (Zabbix e Grafana), detecção de intrusão (Suricata), SIEM (Wazuh) e automação de

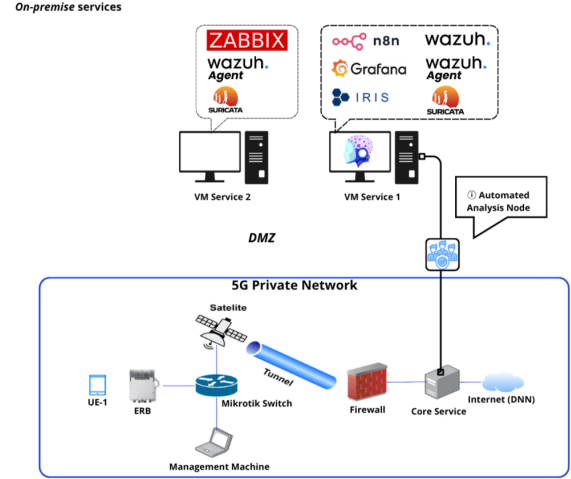


Fig. 1. Arquitetura geral e ambiente experimental detalhado.

fluxos (n8n). O Zabbix coleta métricas dos ativos da rede, enquanto o Grafana, integrado ao banco MariaDB, fornece painéis de visualização para a equipe do Centro de Operações de Rede (NOC).

B. Arquitetura do Framework

O framework foi concebido como uma arquitetura modular para apoio à detecção, análise e resposta a incidentes sobre a infraestrutura descrita anteriormente. Seu fluxo geral, mostrado na Figura 2, é descrito por cinco etapas operacionais: alerta, coleta, normalização, análise por IA e bloqueio/notificação. Funcionalmente, essas etapas são agrupadas em quatro camadas: detecção, orquestração, inteligência e resposta.

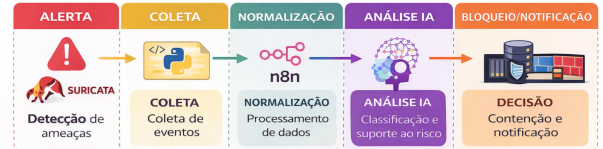


Fig. 2. Fluxo do framework proposto.

Na camada de detecção, o Suricata monitora o tráfego de rede e gera alertas associados a padrões suspeitos, enquanto o Wazuh coleta eventos de host e logs, complementando a caracterização do incidente. Na camada de orquestração, esses eventos são capturados e encaminhados via webhook ao n8n, que implementa um fluxo de resposta a incidentes baseado em múltiplos agentes de IA.

A cadeia de agentes compreende: *Tech Diag*, responsável por analisar os logs com base em regras definidas em seu *System Message* e gerar um diagnóstico estruturado em JSON; *SOC Guardian*, que converte o diagnóstico técnico em um alerta objetivo para WhatsApp; *Decision Validator AI*, que interpreta a resposta do gestor e a classifica como APROVADA ou REJEITADA; e *AI Agent Response*, que executa a mitigação após consultar a *Vector Store* e a lista *blocked_ips*, acionando ferramentas via *Model Context Protocol* (MCP), como *block_ip*, para aplicação da regra no *firewall* [12].

O sistema se baseia em uma lógica operacional quase determinística, guiada por regras explícitas (*System Messages*)

publicadas no repositório do framework [12]. Em termos de engenharia de software para agentes, a arquitetura adota um padrão compatível com *ReAct* (*reasoning and acting*), no qual o modelo alterna interpretação do contexto, consulta à memória e às ferramentas, e emissão de uma ação final. A memória vetorial permite recuperar ocorrências históricas semanticamente correlatas, enquanto a memória operacional local preserva o estado dos bloqueios já aplicados, favorecendo a recuperação de contexto pelo agente de execução.

III. IMPLEMENTAÇÃO DE SEGURANÇA

A implementação do framework de segurança ocorreu de forma gradual, garantindo visibilidade completa do tráfego antes da automação das respostas. O Suricata foi implantado como sensor *network intrusion detection system* (NIDS), com o arquivo *suricata.yaml* configurado para cobrir as redes internas e regras atualizadas via *suricata-update*. O Wazuh atuou como SIEM e *host intrusion detection system* (HIDS), instalado no modelo *all-in-one* (Manager, Indexer e Dashboard).

Para unificar a detecção, os eventos de rede registrados pelo Suricata em *eve.json* foram injetados no Wazuh Agent via diretiva `<localfile>`, permitindo correlação de alertas e visualização centralizada por meio de um painel de *Threat Hunting*. A automação das respostas segue um fluxo orquestrado pelo n8n e acionado por alertas críticos do Wazuh. O *pipeline* envolve diagnóstico técnico, geração de notificação executiva, validação humana no fluxo e execução da contenção pelo agente *Executor*, com consulta à RAG e uso de ferramentas MCP, como *block_ip*.

Na implementação prática, o workflow do n8n foi segmentado em nós especializados, preservando separação clara entre diagnóstico técnico, comunicação executiva e resposta operacional. A Figura 3 apresenta o fluxo completo do sistema. Na etapa de diagnóstico, o *webhook* alimenta o agente *Tech Diag*, o modelo de linguagem, o *parser* estruturado de saída e os componentes de persistência em *Vector Store*, evidenciando a integração entre processamento semântico, armazenamento vetorial e infraestrutura de resposta a incidentes.

O comportamento do agente de diagnóstico é condicionado por um *System Prompt* especializado, que delimita o contexto operacional da análise. Entre as diretrizes definidas, destacam-se a análise dos logs em janelas de cinco minutos com avanço de um minuto, a identificação dos ativos e portas relevantes da infraestrutura, a interpretação de padrões como ACK/TCP Manipulation como possíveis indícios de ataques de negação de serviço distribuída (DDoS), a avaliação de volumetria com base no campo *fireddtimes*, a diferenciação entre varreduras isoladas e ataques distribuídos e a consideração de persistência temporal dos eventos. Além disso, o agente é instruído a interpretar logs vazios como ausência de evento relevante, reduzindo diagnósticos espúrios.

A etapa subsequente corresponde, na Figura 3, à região *Criação de Relatório Amigável*, na qual o agente *SOC Guardian* converte o diagnóstico técnico em uma comunicação sintética e operacional, encaminhada via WhatsApp ao responsável. Dessa forma, reduz-se o hiato entre a detecção automatizada e a decisão humana sobre a contenção.

No plano de implementação, o gatilho operacional do fluxo é realizado pelo script *ai_webhook_cyber.py*, responsável por consolidar o último minuto de observação, acionar o *webhook* do n8n e encaminhar o sumário do evento suspeito ao agente de diagnóstico. O componente também interage com o artefato *blocked_ips.json*, utilizado como memória operacional local para registrar bloqueios ativos, evitar reexecuções redundantes e complementar a memória histórica semântica provida pela *Vector Store*.

A fase final do workflow corresponde, na Figura 3, à região de *Resposta do CSIRT*, na qual atua o agente de resposta a incidentes de segurança em computadores (CSIRT): o agente executor recebe a decisão validada, consulta o estado do sistema e a memória histórica via RAG, e aciona ferramentas através do MCP, como *status*, *block_ip* e *unblock_ip*. O servidor MCP atua como camada de mediação entre a LLM e os atuadores do ambiente, preservando auditabilidade, semântica explícita das ações e governança da contenção aplicada. Quando o bloqueio é recomendado, o subsistema Python executa a regra *iptables -A INPUT -s <IP_SUSPEITO> -j DROP*, atualiza a lista de IPs bloqueados e envia notificações via WhatsApp à equipe de segurança em tempo hábil.

IV. VALIDAÇÃO E RESULTADOS EXPERIMENTAIS

O framework foi validado em um cenário de ataque controlado, composto por uma ERB Baicells, UEs registrados e acesso externo viabilizado pelo enlace LEO. A metodologia de avaliação contemplou duas fases distintas de intrusão, originadas do endereço IP 10.7.101.202 (instância do Grafana) contra o alvo 10.7.101.206 (servidor Zabbix).

A. Fase de Reconhecimento (Scanning)

Na etapa inicial, executou-se uma varredura de portas com a ferramenta Nmap para mapeamento da superfície de ataque, identificando serviços ativos nas portas 22 (SSH), 80 (HTTP) e 3306 (MySQL). O comando utilizado na coleta foi registrado no ambiente experimental, dado por: `nmap -sS -sV -O -T4 10.7.101.206`.

O Suricata reconheceu prontamente a assinatura do *User-Agent* do Nmap e padrões de identificação do sistema operacional (*fingerprinting*) associados à varredura realizada. Os eventos processados pelo Wazuh, mostrados na Figura 4, confirmaram a visibilidade do tráfego anômalo e a eficácia da detecção precoce.

B. Fase de Exploração e Mitigação (DoS)

Na segunda fase, simulou-se um ataque de DoS na camada de aplicação, por meio da injeção massiva de requisições contra o servidor web do alvo. O Wazuh detectou o comportamento e disparou alertas críticos referentes à alta taxa de requisições e aos sucessivos erros HTTP 400.

O IDS Suricata, atuando em conjunto com o SIEM, reportou um volume anômalo de acessos, indicando alta taxa de requisições HTTP POST e repetição excessiva de cabeçalhos, características compatíveis com o ataque conduzido.

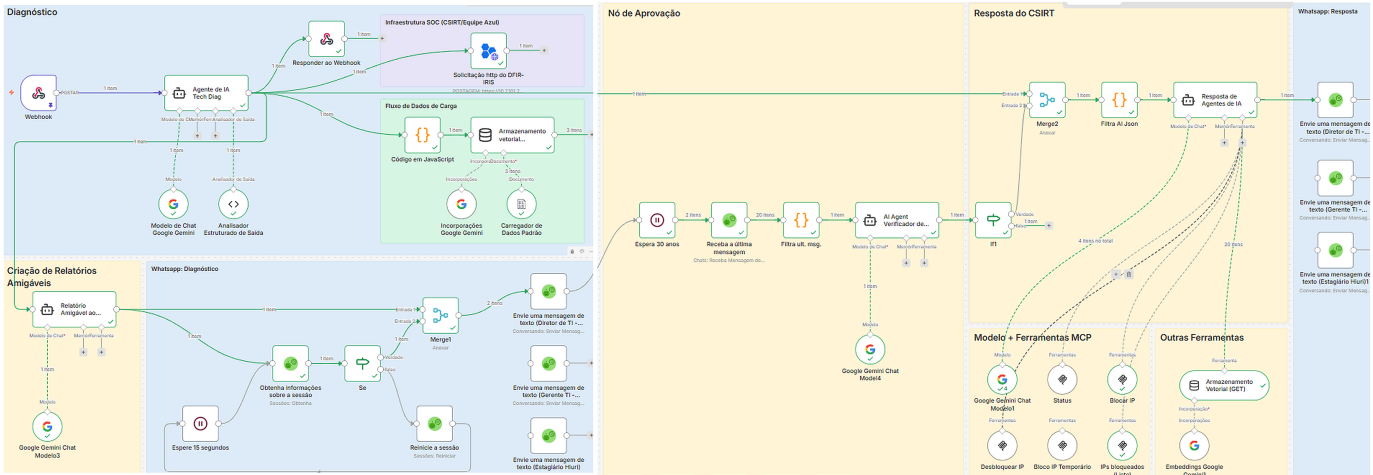


Fig. 3. Fluxo completo do sistema implementado no n8n.

timestamp	agent_name	rule.description
Feb 2, 2026 @ 15:08:06.457	Grupo2-grafana	Suricata: Alert - ET SCAN Possible Nmap User-Agent Observed
Feb 2, 2026 @ 15:08:06.456	Grupo2-grafana	Suricata: Alert - ET SCAN Possible Nmap User-Agent Observed
Feb 2, 2026 @ 15:08:06.456	Grupo2-grafana	Suricata: Alert - ET SCAN Possible Nmap User-Agent Observed
Feb 2, 2026 @ 15:08:06.455	Grupo2-grafana	Suricata: Alert - ET SCAN Possible Nmap User-Agent Observed

Fig. 4. Detecção do evento de varredura Nmap pelo Wazuh.

Após o reconhecimento da ameaça, o sistema isolou a origem do tráfego malicioso por meio da análise automatizada dos logs, acionando o webhook para o n8n iniciar o fluxo de resposta e contenção. Os alertas correlacionados reportados durante o ataque são ilustrados na Figura 5.

timestamp	agent_name	rule.description
Feb 2, 2026 @ 15:08:03.542	Grupo2-grafana	Suricata: Alert - SURICATA ICMPv4 unknown code
Feb 2, 2026 @ 15:08:03.541	Grupo2-grafana	Suricata: Alert - SURICATA ICMPv4 unknown code
Feb 2, 2026 @ 15:07:59.537	Grupo2-grafana	Suricata: Alert - SURICATA ICMPv4 unknown code
Feb 2, 2026 @ 15:07:59.537	Grupo2-grafana	Suricata: Alert - SURICATA ICMPv4 unknown code
Feb 2, 2026 @ 15:07:57.536	Grupo2-grafana	Suricata: Alert - SURICATA ICMPv4 unknown code
Feb 2, 2026 @ 15:07:57.536	Grupo2-grafana	Suricata: Alert - SURICATA ICMPv4 unknown code
Feb 2, 2026 @ 15:07:55.534	Grupo2-grafana	Suricata: Alert - SURICATA ICMPv4 unknown code
Feb 2, 2026 @ 15:07:55.534	Grupo2-grafana	Suricata: Alert - SURICATA ICMPv4 unknown code

Fig. 5. Eventos e alertas reportados durante o ciberataque DoS.

C. Exemplo de Log Processado pelo Script de IA

Como evidência de execução real do fluxo automatizado, a Figura 6 apresenta um trecho do log operacional produzido pelo script Python ai_webhook_cyber.py, responsável por integrar coleta local, sumarização de eventos e disparo do webhook para análise via IA. Observa-se, inicialmente, a criação do cache local e, na execução subsequente, a detecção de 13.604 novas linhas, seguida da consolidação de um padrão de ACK/TCP Manipulation com 5.664 ocorrências no intervalo observado. Esse comportamento indica que o pré-processamento não se limita à retransmissão do log bruto, realizando também agregação temporal, extração do par atacante–alvo e organização dos eventos em um formato adequado ao consumo pelos agentes do workflow. Como o bloco de eventos brutos é altamente repetitivo, o trecho exibido foi condensado para destacar a estrutura recorrente observada pelo agente Wazuh a partir dos alertas oriundos

do Suricata, preservando os campos centrais de inspeção, incluindo timestamp, rule.level, rule.description e rule.firedtimes.

```
Bloco 1 -- Inicializacao
Execucao: 2026-03-01 22:18:51
Cache inicial criado, com proxima coleta agendada.
Sem novas entradas no ultimo minuto.

Bloco 2 -- Coleta incremental
Execucao: 2026-03-01 22:19:51
Novas linhas detectadas: 13.604

Bloco 3 -- Sumario do evento suspeito
Tipo: ACK / TCP Manipulation
Atacante: 10.7.101.206
Alvo: 10.7.101.202:3000
Ocorrencias consolidadas: 5.664

Bloco 4 -- Estrutura dos eventos brutos
Cabecalho recorrente: [ACK ATTACK EVENTS]
Campos preservados: timestamp, rule.level, rule.description e rule.firedtimes
Exemplo esquematico:
{"timestamp": "...", "rule":
{"level": 3, "description": "Suricata:
Alert", "firedtimes": 5664, ...}}
Repeticao: multiplos eventos Wazuh/Suricata no intervalo
analizado
```

Fig. 6. Trecho do log operacional de log.txt.

A execução integrada de todo o workflow é apresentada na Figura 3. Nela, observa-se a articulação completa entre os módulos de diagnóstico, comunicação, nó de aprovação e resposta CSIRT, evidenciando a passagem de contexto entre os agentes e a coordenação ponta a ponta do processo de mitigação. Essa visão consolidada reforça que a arquitetura proposta não corresponde a um protótipo conceitual isolado, mas a uma implementação operacional com encadeamento explícito entre inferência, mediação humana e execução de resposta.

Como evidência da efetivação da contenção, a Figura 7 mostra a validação da regra de bloqueio no firewall após a execução do workflow. O registro visual da cadeia INPUT demonstra que a ação recomendada pela IA foi materializada em uma regra de descarte para o endereço atacante, o que confirma o fechamento do ciclo entre detecção, decisão e execução. Sob a ótica experimental, essa etapa é decisiva para

demonstrar que a resposta produzida pelo sistema transcende a geração de alertas e alcança intervenção técnica verificável sobre a infraestrutura de rede.

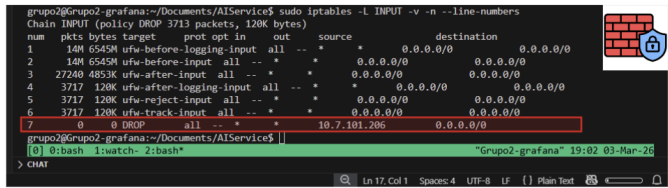


Fig. 7. Validação da regra de bloqueio de IP no firewall após a execução automatizada do workflow gerada pela IA.

A partir do acionamento dos alertas, a orquestração via n8n demonstrou, neste cenário experimental, cobertura das táticas TA0043 (Reconnaissance) e TA0040 (Impact) da matriz MITRE ATT&CK [11], abrangendo a varredura da superfície de ataque e o evento de negação de serviço analisado, conforme o mapeamento das regras de detecção do Wazuh, com 28 alertas correlacionados em 24h, usando a técnica de RAG com *vector store*. No mesmo recorte, o IP atacante foi inserido na regra DROP do firewall em até 1 minuto, enquanto o LLM interpretou o evento e encaminhou, de forma imediata, um relatório via WhatsApp à equipe de resposta, como ilustrado na Figura 8. Essas métricas referem-se à execução considerada neste estudo e devem ser interpretadas como evidências experimentais da viabilidade operacional do pipeline proposto. Com isso, os resultados obtidos cobrem uma validação funcional do pipeline, enquanto a caracterização completa de desempenho, incluindo precisão, *recall*, taxa de falsos positivos, latência e impacto no tráfego legítimo durante a contenção, demandaria um conjunto mais amplo e diversificado de cenários de ataque e condições de tráfego de fundo, constituindo objeto de investigação futura.

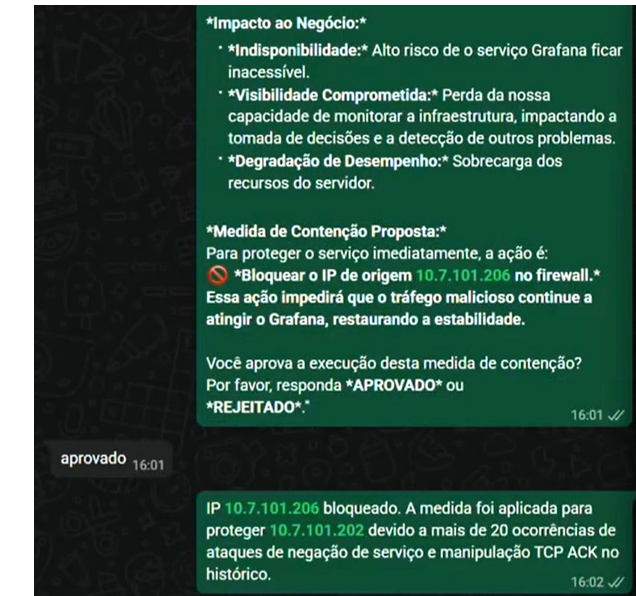


Fig. 8. Exemplo de relatório encaminhado via WhatsApp à equipe de resposta após a interpretação do incidente pelo LLM.

V. CONCLUSÕES

Este trabalho demonstrou a viabilidade de um framework de IA para segurança cibernética em redes móveis privativas com RAN remota via enlace LEO, integrando Wazuh, Suricata, scripts em Python, n8n, memória vetorial e ferramentas MCP em um ciclo auditável de diagnóstico, comunicação executiva, validação humana e resposta semiautomatizada. Os resultados obtidos indicaram capacidade efetiva de apoiar a detecção, a análise contextual e a mitigação de eventos de *scanning* e negação de serviço em ambiente controlado. Como principais contribuições deste trabalho, destacam-se a proposição de uma arquitetura híbrida para detecção e resposta assistidas por IA, a integração entre LLM, RAG, n8n e MCP, e a validação prática do *pipeline* completo com evidências de detecção, decisão e contenção. Em conjunto, os resultados reforçam o potencial da IA como elemento de apoio à operação de segurança em redes móveis não públicas, ampliando a capacidade de correlação, interpretação e resposta a incidentes em infraestruturas distribuídas. Cabe observar que a validação não incluiu comparação experimental direta com abordagens puramente baseadas em regras, sendo prevista em trabalhos futuros.

VI. TRABALHOS FUTUROS

Como trabalhos futuros pretende-se: (i) expandir o framework para arquitetura multiagente distribuída, com contenção cooperativa; (ii) aprimorar as decisões de resposta com adaptação dinâmica de políticas e isolamento progressivo de sessões comprometidas; (iii) investigar defesa autônoma para redes 5G/6G privativas, combinando memória contextual, políticas e execução distribuída; (iv) análise quantitativa e qualitativa comparativa com modelos baseline.

REFERÊNCIAS

[1] 3GPP, “Service requirements for the 5G system (3GPP TS 22.261),” ETSI TS 122 261 V19.12.0 (Release 19), Oct. 2025.

[2] Anatel, “Redes Privativas,” página institucional. Acesso em: 20 mar. 2026.

[3] J. M. O. das Mercês et al., “Core 5G Open Source com RAN Comercial: Uma Avaliação de Desempenho em Rede 5G Privativa,” in XLIII SBt, Natal, RN, Brasil, Sep.–Oct. 2025.

[4] ETSI, “Security architecture and procedures for 5G System (3GPP TS 33.501),” ETSI TS 133 501 V18.9.0 (Release 18), Apr. 2025.

[5] S. Tariq et al., “A survey on 5G private and B5G network threats and safeguarding AI-based security mechanisms through the layered analysis,” *Computer Networks*, vol. 271, Oct. 2025.

[6] Wazuh Documentation, “Network IDS integration (integrate Suricata with Wazuh),” Proof of Concept guide.

[7] Wazuh Documentation, “Getting started with Wazuh,” documentação oficial. [Online]. Available: <https://documentation.wazuh.com/current/getting-started/index.html>. Acesso em: 14 abr. 2026.

[8] Open Information Security Foundation, “Suricata User Guide,” documentação oficial. [Online]. Available: <https://docs.suricata.io/>. Acesso em: 14 abr. 2026.

[9] Y. Chen et al., “A survey of large language models for cyber threat detection,” *Computers & Security*, vol. 145, Oct. 2024.

[10] NIST, A. Nelson et al., “Incident Response Recommendations and Considerations for Cybersecurity Risk Management: A CSF 2.0 Community Profile,” NIST SP 800-61r3, Apr. 2025.

[11] MITRE, “MITRE ATT&CK®,” base de conhecimento pública sobre táticas e técnicas adversárias. Acesso em: 20 mar. 2026.

[12] G. Franco, G. Baltazar, H. Noronha, M. Tomita, C. Callogero, “System Messages dos agentes TechDiag, SOC Guardian, Decision Validator e Executor,” MobNet-AICyberSec, GitHub repository, Apr. 7, 2026. [Online]. Available: <https://github.com/NoronhaH/MobNet-AICyberSec>. Accessed on: Apr. 7, 2026.