PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS
ESCOLA POLITÉCNICA E DE ARTES
GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO

ANÁLISE DE VIABILIDADE E DESEMPENHO DO RASPBERRY PI COMO FIREWALL DE BAIXO CUSTO PARA REDES DE PEQUENO PORTE

JEFFERSON RUAN TELES DE OLIVEIRA

Goiânia
2025

JEFFERSON RUAN TELES DE OLIVEIRA

ANÁLISE DE VIABILIDADE E DESEMPENHO DO RASPBERRY PI COMO FIREWALL DE BAIXO CUSTO PARA REDES DE PEQUENO PORTE

Trabalho de Conclusão de Curso apresentado à Escola Politécnica e de Artes, da Pontifícia Universidade Católica de Goiás, como parte dos requisitos para a obtenção do título de Bacharel em Engenharia de Computação.

Orientador (a): Prof. M.E.E. Marcelo Antonio Adad de Araújo.

Goiânia

2025

JEFFERSON RUAN TELES DE OLIVEIRA

ANÁLISE DE VIABILIDADE E DESEMPENHO DO RASPBERRY PI COMO FIREWALL DE BAIXO CUSTO PARA REDES DE PEQUENO PORTE

Este trabalho de Conclusão de Curso julgado adequado para obtenção do título de Bacharel em Engenharia de Computação, e aprovado em sua forma final pela Escola Politécnica e de Artes, da Pontifícia Universidade Católica de Goiás, em

_____/_____/_____.

Prof. Dr Luiz Álvaro de Oliveira Junior

Coordenador de Trabalho de Conclusão de Curso.

Banca examinadora:

Orientador: Prof. Marcelo Antonio Adad de Araújo, M.E.E.

Prof. Carlos Alexandre Ferreira de Lima, M.E.E.

Prof. Nilson Cardoso Amaral, PhD.

Goiânia

2025

RESUMO

A crescente dependência de sistemas informatizados por empresas e redes residenciais torna a segurança de redes um componente indispensável, mas soluções sofisticadas frequentemente são inacessíveis para pequenas empresas e ambientes com orçamentos limitados. Diante desse cenário, este trabalho investiga a viabilidade técnica do uso do Raspberry Pi como um firewall de baixo custo para redes de pequeno porte. O estudo aborda o problema central de determinar se o Raspberry Pi pode ser uma alternativa funcional e segura a firewalls tradicionais. Para isso, são realizados testes controlados para analisar seu desempenho, segurança, consumo de recursos e estabilidade. Os resultados demonstram a viabilidade técnica do Raspberry Pi como firewall de baixo custo, apresentando desempenho satisfatório e estabilidade em cenários otimizados, além de um consumo de energia significativamente inferior às soluções tradicionais. O estudo busca contribuir para a democratização da cibersegurança, oferecendo subsídios técnicos para a adoção consciente dessa tecnologia em ambientes com recursos limitados.

Palavras-chave: Raspberry Pi; Firewall; Segurança de Redes; Baixo Custo; Desempenho.

ABSTRACT

The increasing reliance on computerized systems by businesses and residential networks makes network security an indispensable component, yet advanced solutions are often inaccessible to small organizations with limited budgets. This work evaluated the technical feasibility of using a Raspberry Pi as a low-cost firewall for small networks. The study examined whether the device could operate as a functional and secure alternative to traditional commercial firewalls. Controlled experiments were carried out to assess performance, security, resource usage, and stability. The results demonstrated that the Raspberry Pi can serve as a viable low-cost firewall, delivering stable performance in optimized scenarios and consuming significantly less power than conventional appliances. The work contributes to the broader adoption of accessible cybersecurity solutions by providing practical insights for environments with restricted resources.

Keywords: Raspberry Pi; Firewall; Network Security; Low Cost; Performance.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS E ABREVIATURAS

ACLs	Access Control Lists
ARM	Advanced RISC Machine
APIs	Application Programming Interfaces
BIP	Binary Integer Program
CIA	Confidentiality, Integrity, and Availability
CPU	Central Processing Unit
DDoS	Distributed Denial of Service
DHCP	Dynamic Host Configuration Protocol
DMZ	Demilitarized Zone
DPI	Deep Packet Inspection
DoS	Denial of Service
Gbps	Gibabites per second
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection Systems
IoT	Internet of Things
IETF	Internet Engineering Task Force
IP	Internet Protocol
IPS	Intrusion Prevention System
IPSec	Internet Protocol Security
LAN	Local Area Network
LCD	Liquid Crystal Display
LGPD	Lei Geral de Proteção de Dados
Mbps	Megabits per second
MEI	Individual Microentrepreneur
MPEs	Micro and Small Businesses
MTT	Mean Time to Recovery
MTU	Maximum Transmission Unit
NAT	Network Address Translation
NGFW	Next-Generation Firewall
ORO	Optimal Rule Ordering Problem
OS	Operating System
PABX	Private Branch Exchange

PIB	Produto Interno Bruto
PMEs	Pequenas e Médias Empresas
RAM	Random Access Memory
RFCs	Request for Comments
RTT	Round Trip Time
SBCs	Single Board Computers
SSH	Secure Shell
SIP	Session Initiation Protocol
SNMP	Simple Network Management Protocol
TCP	Transmission Control Protocol
TI	Tecnologia da Informação
UDP	User Datagram Protocol
UFW	Uncomplicated Firewall
VLAN	Virtual Local Area Network
VoIP	Voice over IP
VPN	Virtual Private Networks
WAN	Wide Area Network

SUMÁRIO

1 INTRODUÇÃO

A crescente dependência de sistemas informatizados por empresas de todos os portes tornou a segurança de redes um componente essencial para a proteção de dados e continuidade dos negócios. Nesse cenário, enquanto grandes corporações dispõem de recursos para investir em soluções sofisticadas de segurança cibernética, pequenas empresas e redes residenciais enfrentam desafios significativos devido a orçamentos limitados e infraestrutura reduzida.

Nesse contexto, torna-se relevante investigar alternativas acessíveis e eficientes para proteger redes de pequeno porte. Uma das soluções é o uso do Raspberry Pi, um microcomputador de baixo custo e consumo energético, que pode ser configurado para desempenhar funções de segurança, como a de *firewall*. No entanto, a viabilidade técnica dessa aplicação ainda precisa ser analisada de forma sistemática.

Dentre os principais mecanismos de defesa em redes, destaca-se o *firewall*, responsável por controlar o tráfego de dados e bloquear acessos não autorizados. No entanto, os dispositivos dedicados a essa função frequentemente possuem custos elevados, o que os torna inacessíveis para muitos pequenos negócios. Assim, surge o problema central deste trabalho: é viável utilizar o Raspberry Pi como uma alternativa funcional e segura a *firewall*s tradicionais em redes de pequeno porte?

Diante dessa questão, este estudo tem como objetivo principal avaliar a viabilidade técnica e econômica do uso do Raspberry Pi como *firewall*, considerando seu desempenho em diferentes cenários de uso. Para isso, foi adotada uma abordagem experimental, que incluiu a execução de testes controlados, análise de métricas de desempenho, como *throughput*, uso de *Central Processing Unit* (CPU), memória e temperatura, a avaliação da resposta a ataques simulados, além da comparação com soluções comerciais e alternativas de *software*. A pesquisa também aborda a estabilidade operacional do dispositivo em uso contínuo.

A relevância deste estudo reside na possibilidade de tornar a segurança de redes mais acessível, especialmente para organizações com poucos recursos. Ao explorar o uso de uma plataforma de baixo custo e alto potencial como o Raspberry Pi, este trabalho busca contribuir com alternativas práticas e sustentáveis para a proteção de ambientes computacionais modestos, incentivando o uso de tecnologias

abertas e a disseminação do conhecimento técnico. Com isso, espera-se contribuir para o debate sobre o uso de tecnologias acessíveis na proteção de redes, especialmente em ambientes que carecem de recursos para investir em soluções tradicionais.

1.1 Justificativa

A pesquisa proposta justifica-se pela necessidade de soluções de segurança digital que sejam ao mesmo tempo eficazes, acessíveis e viáveis para contextos com infraestrutura limitada. Mesmo com a proteção de redes sendo amplamente reconhecida como essencial, observa-se uma carência de alternativas práticas e economicamente viáveis para pequenos negócios, residências e instituições que não dispõem de meios para investir em soluções comerciais sofisticadas. Embora a literatura aborde amplamente o funcionamento e a importância dos *firewalls*, há escassez de estudos práticos que avaliem o desempenho de dispositivos de baixo custo, como o Raspberry Pi, em cenários reais de uso como solução de segurança.

Ao investigar a aplicação do Raspberry Pi como *firewall*, este trabalho pretende suprir uma lacuna ao fornecer uma análise experimental detalhada, com dados concretos sobre desempenho, estabilidade e capacidade de resposta a ameaças.

Além disso, a proposta ajuda a tornar a segurança digital mais acessível, permitindo que pequenas empresas, residências e instituições adotem práticas de proteção que antes eram inviáveis por questões de custo ou complexidade. Ao combinar conhecimento técnico com soluções práticas e de baixo custo, o estudo amplia o acesso a ferramentas de defesa digital e entrega um benefício real tanto para a área acadêmica quanto para a sociedade.

1.2 Objetivos

Após a justificativa apresentada, esta seção descreve os objetivos que orientam a pesquisa. A definição clara desses objetivos direciona o desenvolvimento do trabalho, a coleta e a análise dos dados, além da avaliação dos resultados esperados, assegurando que o escopo e as metas do projeto estejam bem delimitados.

1.2.1 Objetivo Geral

Avaliar a viabilidade técnica do uso do Raspberry Pi como *firewall* de baixo custo em redes de pequeno porte, considerando aspectos como desempenho, limitações, consumo de recursos e estabilidade em diferentes cenários de tráfego e ataque.

1.2.2 Objetivos Específicos

- Realizar uma análise do desempenho do Raspberry Pi como *firewall*, utilizando métricas como *throughput*, uso de CPU, memória e temperatura em diferentes condições de carga.
- Investigar a capacidade do dispositivo de lidar com conexões simultâneas e regras de *firewall* de diferentes complexidades.
- Simular cenários de ataque para avaliar a resposta e resiliência do Raspberry Pi frente a ameaças comuns.
- Comparar o desempenho do Raspberry Pi com soluções comerciais.
- Avaliar a estabilidade do Raspberry Pi em operação contínua e sua escalabilidade em ambientes com crescimento gradual da rede.

1.3 Metodologia

Quanto à sua natureza, esta pesquisa é aplicada, pois avalia, na prática, a viabilidade técnica de uma solução de *firewall* utilizando *hardware* de baixo custo, com foco no Raspberry Pi. Busca-se, com isso, propor uma alternativa viável e acessível para ambientes de rede com recursos limitados.

Quanto aos objetivos, trata-se de uma pesquisa exploratória e descritiva. A abordagem exploratória foi utilizada para investigar as potencialidades e limitações do uso do Raspberry Pi como *firewall*. Já a abordagem descritiva serviu para apresentar os resultados obtidos durante os testes realizados, fornecendo uma visão detalhada sobre o desempenho, consumo de recursos e eficácia da solução proposta.

Com relação aos procedimentos técnicos, esta pesquisa é bibliográfica e experimental. A pesquisa bibliográfica tem como objetivo levantar e analisar

informações relevantes sobre *firewalls*, segurança de redes e o uso do Raspberry Pi em soluções de segurança, utilizando como fontes livros, artigos científicos, trabalhos de conclusão de curso, dissertações e teses disponíveis em bases de dados.

Já a pesquisa experimental foi conduzida por meio da implementação prática de um ambiente de testes, onde o Raspberry Pi foi configurado como um *firewall*, possibilitando a análise empírica de seu desempenho e funcionalidade.

1.4 Resultados Obtidos

Os resultados alcançados neste trabalho confirmaram a viabilidade técnica do uso do Raspberry Pi como uma solução de *firewall* de baixo custo para redes de pequeno porte, demonstrando desempenho consistente em ambientes que possuem limitações de recursos, como pequenas empresas, residências e laboratórios. De forma geral, verificou-se que:

- O Raspberry Pi apresentou throughput adequado, baixa latência e boa estabilidade operacional, desde que configurado corretamente e utilizando um conjunto de regras otimizado.
- O dispositivo foi capaz de suportar um volume significativo de conexões simultâneas e tráfego de rede, operando dentro dos limites impostos por seu hardware, sem interrupções críticas.
- Nos cenários simulados de ataque, o firewall baseado no Raspberry Pi manteve-se estável e respondeu de forma eficiente às tentativas de intrusão, evidenciando sua capacidade de atuação em situações reais.

Além disso, os testes evidenciaram que, embora o Raspberry Pi não substitua dispositivos corporativos de alta performance, ele se apresenta como uma opção plenamente utilizável em cenários de menor escala. Os dados obtidos fornecem subsídios concretos para que administradores de rede e profissionais de Tecnologia da Informação (TI) avaliem essa tecnologia com segurança e embasamento técnico.

Por fim, o estudo contribuiu para o avanço do conhecimento aplicado na área de segurança de redes, demonstrando que plataformas acessíveis, como o Raspberry Pi, podem desempenhar papéis relevantes na proteção de infraestrutura digital, desde que corretamente planejadas, configuradas e monitoradas.

1.5 Organização do Trabalho

Este trabalho está estruturado em sete capítulos, organizados de forma a conduzir o leitor desde os fundamentos teóricos até a análise dos resultados experimentais e as considerações finais.

O Capítulo 1 apresenta o tema, a justificativa, os objetivos, a metodologia adotada e os resultados obtidos, contextualizando a relevância da pesquisa e delimitando seu escopo.

O Capítulo 2 reúne a fundamentação teórica necessária para o entendimento do estudo, abordando conceitos essenciais de segurança de redes, funcionamento de firewalls, desafios enfrentados por pequenas empresas e as características do Raspberry Pi como plataforma de baixo custo para aplicações de segurança.

O Capítulo 3 descreve as métricas utilizadas para avaliar o desempenho, o consumo de recursos, a estabilidade e a segurança da solução proposta, estabelecendo os critérios pelos quais o *firewall* foi analisado.

O Capítulo 4 detalha o ambiente de implementação, incluindo a configuração do Raspberry Pi, a infraestrutura de rede, os softwares utilizados, as políticas aplicadas e os componentes que compõem o *firewall*.

O Capítulo 5 apresenta o planejamento e a execução dos testes experimentais, demonstrando os procedimentos adotados, os cenários construídos e as ferramentas empregadas na coleta dos dados.

O Capítulo 6 discute e interpreta os resultados obtidos nos testes, relacionando-os às métricas definidas e avaliando a viabilidade da solução, suas limitações e seu comportamento frente aos diferentes cenários propostos.

Por fim, o Capítulo 7 reúne as considerações finais, destacando as contribuições do estudo, as conclusões alcançadas e sugestões para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo tem como propósito estabelecer a base conceitual e teórica que sustenta a presente pesquisa. Serão abordados os principais conceitos e conhecimentos relacionados à segurança de redes, *firewalls*, e o uso de microcomputadores de baixo custo, como o Raspberry Pi, em contextos de segurança cibernética. A revisão bibliográfica aqui apresentada visa fornecer o embasamento necessário para a compreensão dos desafios e das soluções propostas, além de contextualizar o problema de pesquisa no cenário atual da tecnologia da informação.

2.1 Introdução à Segurança de Redes

A segurança da informação tem evoluído constantemente, acompanhando o crescimento tecnológico e a interconectividade entre sistemas. Inicialmente, a proteção da informação baseava-se em métodos físicos e administrativos, como o controle rígido de acesso a documentos sigilosos. No entanto, a introdução dos computadores e redes ampliou os desafios e exigiu o desenvolvimento de ferramentas automatizadas para garantir a proteção de dados digitais (STALLINGS, 2011).

Com o surgimento de ambientes compartilhados e a popularização das redes públicas, tornou-se indispensável adotar mecanismos especializados para impedir acessos não autorizados e preservar a integridade e confidencialidade das informações. Assim, consolidou-se o conceito de segurança computacional, definido como um conjunto de técnicas voltadas à proteção contra ameaças digitais (STALLINGS, 2011). Além disso, a expansão dos sistemas distribuídos e a necessidade de comunicação eficiente entre dispositivos interligados exigiram medidas específicas para proteger dados em trânsito. Como observado por Stallings (2011), esse contexto levou ao desenvolvimento da segurança de redes, que não se limita apenas à proteção de dispositivos individuais, mas também abrange as conexões que sustentam a infraestrutura digital das organizações. Dada a complexidade das redes modernas, o termo segurança da internet passou a ser empregado para descrever as estratégias de defesa utilizadas nesse ambiente interconectado.

É importante ressaltar que a distinção entre segurança computacional e segurança de redes não é absoluta, pois muitas ameaças transitam entre esses dois domínios. Por exemplo, os vírus de computador podem ser disseminados tanto por mídias removíveis quanto via Internet, exigindo a aplicação de mecanismos combinados de proteção. Assim, a segurança de redes se integra a um sistema mais amplo de defesa da informação, contemplando não apenas a prevenção de ataques, mas também a resposta e recuperação diante de ameaças (STALLINGS, 2011). Reconhecer essa interdependência é necessário para que as estratégias de defesa adotadas no ambiente cibernético atual sejam de fato completas e eficazes.

2.1.1 Tríade Confidencialidade, Integridade e Disponibilidade

A Tríade CID, ilustrada na figura 1, é um pilar estrutural da segurança da informação, engloba três princípios essenciais: Confidencialidade, Integridade e Disponibilidade. Esses princípios são a base para proteger sistemas de informação e redes contra uma variedade de ameaças. Dependendo do contexto, a relevância de cada princípio pode variar, mas todos são indispensáveis para assegurar a segurança de um sistema.

Figura 1 - Tríade CID

Fonte: Pronnus, 2025

A Confidencialidade, conforme definido por Cole (2009), refere-se à proteção das informações contra a divulgação não autorizada. Essa violação pode ocorrer tanto de maneira intencional, como no caso de um atacante que quebra um sistema de criptografia para acessar dados sensíveis, quanto de maneira não intencional, por meio de falhas humanas ou problemas sistêmicos. A proteção da confidencialidade é crítica em contextos como o de agências governamentais ou empresas que lidam com informações sensíveis, pois o acesso não autorizado pode resultar em sérios danos à privacidade e segurança de indivíduos e organizações.

A Integridade busca assegurar que as informações não sejam alteradas sem a devida autorização. Cole (2009) descreve três aspectos essenciais para garantir a integridade das informações: a prevenção de modificações não autorizadas por usuários, a prevenção de alterações não intencionais por usuários autorizados e a preservação da consistência interna e externa dos dados. A integridade é vital em ambientes onde a precisão e a consistência das informações são cruciais, como em sistemas bancários e bases de dados organizacionais, onde qualquer modificação errônea pode comprometer a funcionalidade e a confiança no sistema.

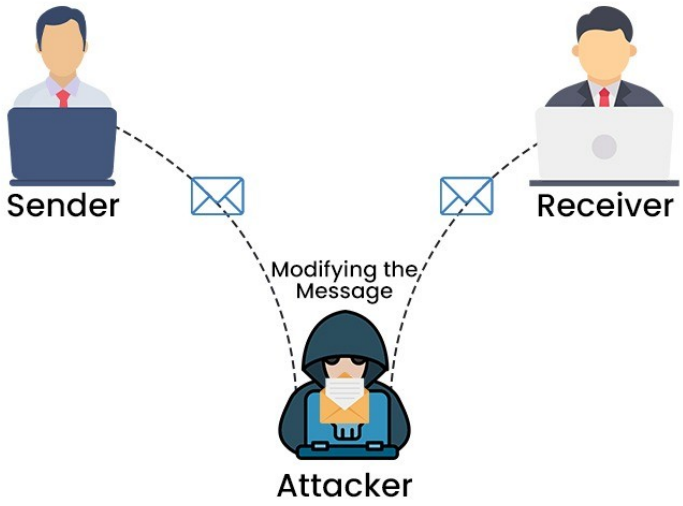
A Disponibilidade assegura que as informações e os sistemas estejam acessíveis quando necessário pelos usuários autorizados. Sem disponibilidade, mesmo que as informações sejam confidenciais e íntegras, elas se tornam inúteis se não puderem ser acessadas no momento certo. Em muitos sistemas críticos, como em servidores financeiros, hospitais, sistemas de controle de tráfego aéreo ou infraestruturas de energia, a falta de disponibilidade pode resultar em consequências graves, prejudicando a operação dos serviços e comprometendo a confiança do usuário no sistema.

Assim, os princípios da Tríade CID, Confidencialidade, Integridade e Disponibilidade, são interdependentes e essenciais para a segurança de redes. A eficácia de um sistema de segurança depende do equilíbrio entre esses três pilares. Falhas em um desses aspectos podem comprometer todo o sistema, tornando vulneráveis os dados e operações. Portanto, é determinante que os administradores de rede implementem medidas adequadas para proteger cada um desses princípios conforme as necessidades específicas do sistema. Depois de compreender os pilares da segurança, é necessário examinar as ameaças que podem comprometer a Tríade CID, as quais se apresentam de diferentes maneiras, como nos ataques ativos e passivos.

2.1.2 Classificação de Ataques em Redes

Os ataques ativos e passivos são técnicas fundamentais no contexto de segurança cibernética, cada uma com características e objetivos distintos, mas frequentemente complementares. Enquanto os ataques ativos visam comprometer diretamente os sistemas ou recursos da vítima, os ataques passivos, por sua vez, envolvem a coleta de informações sem interação direta, o que pode tornar esses ataques, em certos contextos, mais eficazes. No entanto, o valor de cada um se potencializa quando utilizado em conjunto, formando uma abordagem estratégica para a invasão de sistemas. No caso dos ataques ativos, o atacante busca uma interação direta com o sistema alvo, conforme ilustrado na figura 2, como no exemplo de um ataque de Negação de Serviço (DoS), onde o objetivo é impedir que usuários legítimos acessem um recurso.

Figura 2 - Ataque Ativo

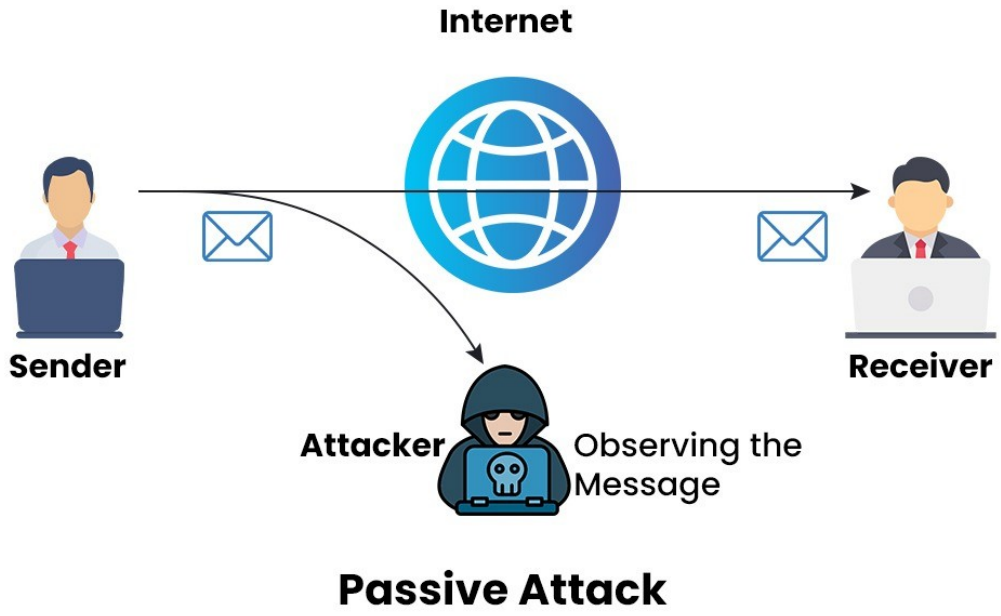


Fonte: PyNetLabs, 2025

Um ataque de DoS pode ser tão simples quanto cortar o acesso de uma pessoa a um servidor, prejudicando sua capacidade de realizar seu trabalho. De acordo com Cole (2009), esses ataques são relativamente fáceis de realizar, principalmente devido à disponibilidade de ferramentas especializadas e à facilidade de execução, já que um atacante pode lançar um DoS sem a necessidade de

acessar diretamente o sistema alvo. A eficácia de um ataque ativo, no entanto, também depende da infraestrutura da vítima e da vulnerabilidade dos recursos que estão sendo atacados. Por outro lado, os ataques passivos, como a captura de pacotes (*sniffing*), têm como característica a monitoração e o registro de dados que trafegam por uma rede, sem que o atacante precise interagir diretamente com o sistema, conforme ilustrado na figura 3.

Figura 3 - Ataque Passivo



Fonte: PyNetLabs, 2025

Essa abordagem permite que o atacante colete informações valiosas, como credenciais de *login*, sem que a vítima perceba. Como exemplo, um atacante pode utilizar programas específicos para capturar pacotes de autenticação de sistemas Windows e, a partir daí, realizar a quebra da senha armazenada. Cole (2009) destaca que, embora o processo de captura de uma única senha possa parecer trabalhoso, um atacante pode configurar a ferramenta para rodar durante períodos estratégicos, como as manhãs de trabalho, e assim coletar centenas de credenciais em um curto intervalo de tempo.

Além disso, os ataques passivos também são eficazes na coleta de informações externas que podem auxiliar em um ataque ativo posterior. Um exemplo é o ataque de reconhecimento, no qual o atacante observa o comportamento da empresa, como os fornecedores e os produtos entregues, para deduzir os sistemas

operacionais ou as atualizações realizadas pela vítima. Essa coleta de informações aparentemente simples pode fornecer dados suficientes para um ataque direcionado mais eficaz.

Em suma, tanto os ataques ativos quanto passivos desempenham papéis críticos na segurança cibernética. Embora os ataques ativos busquem uma invasão direta, os passivos oferecem uma abordagem mais sutil e, muitas vezes, mais eficaz, ao permitir que o atacante colete informações sem ser detectado. A combinação de ambas as abordagens aumenta a probabilidade de sucesso de um ataque, como se observa em diversas estratégias de invasão. Conclui-se, portanto, que a segurança de qualquer sistema passa necessariamente pela compreensão e proteção contra ambos os tipos de ataques.

2.1.3 Ameaças e Vulnerabilidades Comuns

O avanço da tecnologia e a crescente dependência da Internet tornaram as redes de computadores alvos frequentes de ataques cibernéticos. Esses ataques são motivados por diferentes fatores, como obtenção de vantagens econômicas, fraudes, disputas geopolíticas e a simples busca por desafios técnicos. Independentemente do motivo, o principal objetivo desses ataques é comprometer os princípios fundamentais da segurança da informação: confidencialidade, integridade e disponibilidade dos sistemas e dados (COLE, 2009).

De acordo com Cole (2009), esses ataques podem ser classificados em quatro categorias principais:

1. Ataque de Modificação: Consiste na alteração não autorizada de informações. Esse tipo de ataque pode comprometer a integridade dos dados, afetando decisões e processos baseados em informações adulteradas. Um exemplo clássico é a alteração de registros financeiros para desviar fundos ilicitamente.
2. Ataque de Repudiação: Ocorre quando um usuário nega ter realizado uma determinada ação ou transação. Esse tipo de ataque está diretamente relacionado à falta de mecanismos de autenticação e registro adequados. Em sistemas financeiros e empresariais, a repudiação pode comprometer a confiabilidade das operações, exigindo o uso de assinaturas digitais e *logs* de auditoria para mitigar tais riscos.

3. Ataque de Negação de Serviço (DoS): É uma tática amplamente utilizada para tornar recursos indisponíveis. Ele ocorre quando um sistema é sobrecarregado por requisições maliciosas, impedindo usuários legítimos de acessarem serviços essenciais. Essa técnica é frequentemente explorada em ataques contra servidores web e infraestruturas críticas, afetando empresas e serviços públicos.
4. Ataque de Acesso: Envolve a obtenção não autorizada de informações e recursos de rede. Essa categoria inclui técnicas como exploração de vulnerabilidades, uso de credenciais comprometidas e engenharia social. O acesso não autorizado pode resultar no roubo de dados sensíveis, comprometendo a privacidade e a segurança das informações armazenadas.

A compreensão dos distintos tipos de ataques cibernéticos revela-se um requisito para a elaboração de estratégias de defesa eficientes. Nesse contexto, medidas como criptografia, controle de acesso, monitoramento de redes e resposta a incidentes desempenham um papel central na mitigação dessas ameaças e na proteção dos sistemas e dados contra agentes maliciosos. Para combater adequadamente essas ameaças, é igualmente importante entender os atores por trás delas. A seguir, serão explorados os diferentes perfis e motivações dos atacantes cibernéticos.

2.1.4 Perfis e Motivações de Atacantes

Para desenvolver estratégias de segurança em redes, é necessário entender a fundo os perfis dos atacantes, visto que isso possibilita a identificação de vulnerabilidades específicas e a adoção de medidas de defesa apropriadas.

Anderson (1980) demonstra que os atacantes podem ser agrupados conforme suas origens, intenções e métodos de intrusão. Inicialmente, são destacados três perfis distintos: o mascarador, um agente externo que busca se infiltrar no sistema fingindo ser um usuário autorizado, o usuário interno mal-intencionado, um indivíduo interno que utiliza seu acesso para explorar indevidamente os recursos, e o usuário clandestino, que se caracteriza por assumir controle administrativo do sistema para evitar detecção, podendo ser tanto um invasor externo quanto um funcionário interno. Stallings (2011) amplia essa classificação ao propor uma categorização

contemporânea dos invasores, englobando *hackers* motivados por reconhecimento social, grupos criminosos organizados e ataques internos, o que reflete a complexidade das ameaças cibernéticas no século XXI.

Aprofundando a discussão, cada perfil de atacante apresenta desafios específicos. Por exemplo, o mascarador evidencia a importância de mecanismos robustos de autenticação e monitoramento de acessos, já que sua atuação externa visa simplesmente burlar barreiras de segurança. Em contrapartida, o usuário interno mal-intencionado, por ser um usuário legítimo, exige políticas internas rigorosas (como a aplicação do princípio do menor privilégio e a implementação de registros detalhados, como *logs*) para prevenir o abuso de seus direitos de acesso. Por fim, o usuário clandestino, ao manipular os controles de supervisão, ressalta a necessidade de sistemas de auditoria e de ferramentas avançadas de detecção de intrusões.

Os *hackers* tradicionais, por exemplo, muitas vezes agem por desafio intelectual ou reconhecimento, explorando vulnerabilidades oportunistas, como serviços desprotegidos ou credenciais fracas. Essa motivação os diferencia dos criminosos cibernéticos, que operam em redes organizadas visando lucro, como o roubo de dados financeiros, e agem rapidamente para evitar detecção. Já os ataques internos revelam a criticidade de mecanismos que vão além da tecnologia, como a governança de acesso e a gestão de recursos humanos. Como demonstrado por Stallings (2011), a eficácia das estratégias de segurança está intrinsecamente ligada à compreensão detalhada dos perfis comportamentais dos invasores, o que permite a customização de mecanismos de defesa.

Conclui-se, portanto, que a diversidade dos perfis dos atacantes (abrangendo desde invasores motivados pelo prestígio, como os *hackers* tradicionais, até grupos criminosos organizados e ataques oriundos do ambiente interno) impõe a adoção de estratégias de defesa em múltiplas frentes. A implementação de políticas de acesso restrito, sistemas de monitoramento contínuo e atualizações de segurança constantes revelam-se ações determinantes para a redução dos riscos associados a cada tipo de ameaça, assegurando, assim, a integridade e a continuidade das operações dos sistemas de informação.

2.2 Desafios de Segurança em Pequenas Empresas

Com a evolução da Internet e o aumento do trabalho remoto, as empresas enfrentam desafios em termos de segurança cibernética. Aqueles que utilizam a Internet para comunicação, como e-mails e plataformas de compartilhamento de informações, se tornam alvos frequentes de ataques, como fraudes digitais e *ransomware*. Embora grandes empresas também enfrentem ameaças, pequenas empresas, devido à falta de recursos e infraestrutura adequados, tendem a ser mais suscetíveis a ataques cibernéticos. De acordo com um estudo de 2018, os três maiores desafios para pequenas empresas são a falta de *expertise* interna, as restrições orçamentárias e a falta de compreensão sobre como se proteger contra ataques cibernéticos (CHIDUKWANI et al., 2022). Renaud (2016) explica que os cibercriminosos frequentemente visam pequenas empresas devido à percepção de que essas organizações possuem defesas mais fracas.

2.2.1 Relevância Econômica de Pequenas Empresas

As pequenas empresas desempenham um papel central no desenvolvimento econômico e social, sendo responsáveis por uma parcela significativa da geração de empregos e da movimentação financeira no Brasil. De acordo com a Lei Geral das Microempresas e Empresas de Pequeno Porte, instituída em 2006, são classificadas como microempresas aquelas que possuem receita bruta anual de até R$ 360 mil, enquanto empresas de pequeno porte são aquelas com receita anual entre R$ 360 mil e R$ 4,8 milhões (SEBRAE, 2022).

As micro e pequenas empresas (MPEs) representam 99% dos negócios no Brasil e contribuem significativamente para a economia nacional, sendo responsáveis por aproximadamente 30% do Produto Interno Bruto (PIB) do país (CNN Brasil, 2024). Além disso, são responsáveis pela criação de cerca de 80% das vagas de emprego formal, demonstrando seu impacto direto na geração de oportunidades de trabalho. A relevância dessas empresas também se reflete no alto número de novos negócios abertos anualmente. Em 2023, foram registrados mais de 859 mil novos empreendimentos, sendo a maioria composta por micro e pequenas empresas (CNN Brasil, 2024). Além de sua importância econômica, essas empresas lidam com dados sensíveis que, se comprometidos, podem afetar

negativamente a confiança dos consumidores e a operação da empresa. Kayumbe e Michael (2021) destacam que, ao entender as ameaças cibernéticas, as pequenas empresas podem proteger dados vitais, como informações bancárias e detalhes estratégicos, evitando que *hackers* causem danos após acessar suas redes.

Diante desse contexto, é de suma importância compreender a relevância da segurança da informação para essas empresas, especialmente em relação à proteção de suas redes e dados sensíveis. Pequenos negócios, muitas vezes, não possuem estruturas sólidas de segurança cibernética, tornando-se alvos vulneráveis para ataques. Assim, garantir a implementação de medidas eficazes de segurança em pequenas redes mostra-se imprescindível para a sustentabilidade e crescimento desses empreendimentos. Dado o papel determinante das pequenas empresas na economia, sua proteção cibernética adquire uma importância estratégica, especialmente considerando o crescente número de ataques direcionados a esse setor.

2.2.2 A Importância da Cibersegurança para Pequenos Negócios

A segurança de rede é um aspecto de grande relevância para qualquer organização, independentemente do seu porte. No entanto, Pequenas e Médias Empresas (PMEs) têm se tornado alvos frequentes de ataques cibernéticos devido à falta de investimentos adequados em proteção digital. Segundo um levantamento da IBM Security realizado em 2023, aproximadamente 62% dos ataques cibernéticos foram direcionados a PMEs, evidenciando a vulnerabilidade dessas organizações no cenário digital atual. Esse dado ressalta a necessidade de estratégias mais sólidas de segurança para esse segmento.

Essa vulnerabilidade está fortemente associada à subestimação dos riscos dos crimes eletrônicos e à adoção insuficiente de medidas preventivas eficazes. A crença de que o porte reduz o interesse de criminosos digitais ainda é predominante entre gestores de pequenas empresas, o que contribui para a negligência em segurança cibernética. Os atacantes frequentemente visam empresas menores com a suposição de que esses negócios podem não estar adequadamente preparados para lidar com uma violação de segurança de rede (Saha & Anwar, 2023). Além disso, os empreendedores frequentemente têm medo de abrir negócios devido a preocupações com a segurança cibernética e ao custo associado a isso. Enquanto

as empresas maiores costumam ter os recursos necessários para lidar com a segurança cibernética, as pequenas empresas frequentemente não têm (Paulsen, 2023, *apud* Saha & Anwar, 2023). No entanto, essa percepção equivocada pode levar a impactos financeiros e reputacionais significativos, especialmente quando dados sensíveis de clientes e parceiros são comprometidos. De acordo com um estudo da FecomercioSP (2024), incidentes de segurança podem resultar em prejuízos que vão desde a perda de credibilidade no mercado até sanções regulatórias decorrentes do descumprimento de normas de proteção de dados.

Pesquisas apontam um aumento constante nos ataques direcionados a pequenas empresas. De acordo com uma pesquisa conduzida pelo Instituto Ponemon, 66% das pequenas e médias empresas nos EUA, Reino Unido e Europa presenciaram um ataque cibernético malicioso nos últimos 12 meses. O estudo também revelou que 63% dessas empresas sofreram vazamento de dados em 2019, representando um aumento de quase 10% desde 2017. Esses incidentes não apenas comprometem a segurança das informações corporativas, mas também impõem um alto custo financeiro para recuperação. Segundo os relatórios da Ponemon, em 2019, PMEs gastaram, em média, US$ 1,2 milhão para reparar e restaurar sua infraestrutura e ativos de TI após um ataque, além de mais US$ 1,9 milhão devido às perturbações nas operações regulares. O impacto é tão significativo que, conforme algumas estimativas, cerca de 60% das pequenas e médias empresas fecham as portas em até seis meses após um vazamento de dados.

Apesar de sua relevância e vulnerabilidade, muitas pequenas empresas ainda não percebem a gravidade das ameaças digitais. A ideia de que o porte reduz o interesse dos cibercriminosos ainda é comum entre os gestores de PMEs, o que leva à negligência quanto à segurança. Porém, isso pode ser extremamente prejudicial, como ressaltado por Lagazio et al. (2014), que afirmam que as pequenas empresas precisam adotar melhores práticas de segurança cibernética para se protegerem contra os ataques que se intensificam nesse novo domínio de guerra digital, o ciberespaço. Essa afirmação está em linha com o conceito de que o *ciberespaço* é o “quinto domínio da guerra”, conforme mencionado por Tripathi (2015), onde as ameaças cibernéticas se equiparam aos riscos de outras áreas de conflito.

Diante desse cenário, é imprescindível que pequenas empresas passem a encarar a segurança cibernética como um investimento estratégico, e não como um custo dispensável. As primeiras linhas de defesa contra ataques cibernéticos incluem *firewalls*. O *firewall*, quando devidamente configurado, restringe o acesso não autorizado à rede.

2.2.3 Vulnerabilidades Comuns em Pequenas Empresas

As pequenas empresas enfrentam uma crescente ameaça no ambiente digital, sendo frequentemente alvo de ataques cibernéticos que comprometem a segurança de seus dados e sistemas. De acordo com Mohseni (2022), um dos ataques mais comuns a pequenas e médias empresas (PMEs) envolve golpes por e-mail, que são frequentemente disfarçados em anexos maliciosos. Essas ameaças podem se manifestar de diversas formas, incluindo vírus, *malware*, *phishing* e *ransomware*, colocando em risco não apenas a integridade dos dados, mas também a continuidade dos negócios.

O *ransomware*, por exemplo, tem se tornado uma preocupação crescente para os proprietários de pequenas empresas. Esse tipo de *malware* restringe o acesso aos sistemas da vítima e exige um pagamento para que o acesso seja restaurado. Mohseni (2022) destaca que, muitas vezes, o *ransomware* é disseminado por e-mails com anexos maliciosos ou *links* infectados, aproveitando-se da falta de segurança em dispositivos e redes. A ameaça de ataques de força bruta em sistemas remotos inseguros também é uma realidade para as pequenas empresas que utilizam *desktops* remotos, expondo suas redes a riscos elevados.

Nesse cenário, um *firewall* baseado em Raspberry Pi, embora de baixo custo, pode ser uma primeira linha de defesa contra muitas dessas ameaças externas. Por exemplo, a configuração de regras de filtragem de pacotes (*iptables*) pode bloquear tráfego de *phishing* e *ransomware* que tenta se comunicar com servidores de comando e controle conhecidos. Além disso, a capacidade de monitorar o tráfego de rede pode ajudar a identificar tentativas de varredura de portas ou ataques de força bruta.

A falta de segurança cibernética, como Mohseni (2022) observa, pode levar a desastres econômicos e comprometer dados críticos, como informações bancárias, dados pessoais e financeiros de clientes. As ameaças cibernéticas mais prevalentes

em pequenas empresas incluem *malware*, *phishing* e *ransomware*, que visam, essencialmente, prejudicar ou roubar informações valiosas. Uma vez que um *malware* entra na rede de uma empresa, ele pode causar danos significativos, desde o roubo de dados sensíveis até a paralisação de sistemas vitais, impactando diretamente as operações comerciais.

Entre os diversos tipos de *malware* destacados por Mohseni (2022), os mais comuns incluem vírus, *spyware*, *keyloggers*, *worms* e cavalos de Troia (*Trojans*). Cada um desses *malwares* tem um propósito específico, seja para corromper dados, espionagem ou para permitir o controle remoto do sistema da vítima. O *spyware*, por exemplo, é projetado para monitorar e coletar dados sem o conhecimento do usuário, enquanto os *keyloggers* gravam as teclas digitadas, possibilitando o roubo de informações sensíveis como senhas e dados bancários.

Outro ataque significativo é o *phishing*, que envolve o envio de e-mails maliciosos com o objetivo de enganar o destinatário para que revele informações confidenciais, como credenciais bancárias ou de sistemas. A eficácia desses ataques se dá pela utilização de técnicas de engenharia social, que fazem com que as vítimas confiem em mensagens aparentemente legítimas. Além disso, os *bots* maliciosos também representam um risco crescente, pois são *scripts* automatizados que podem assumir o controle remoto de dispositivos e executar ações prejudiciais sem o conhecimento do usuário.

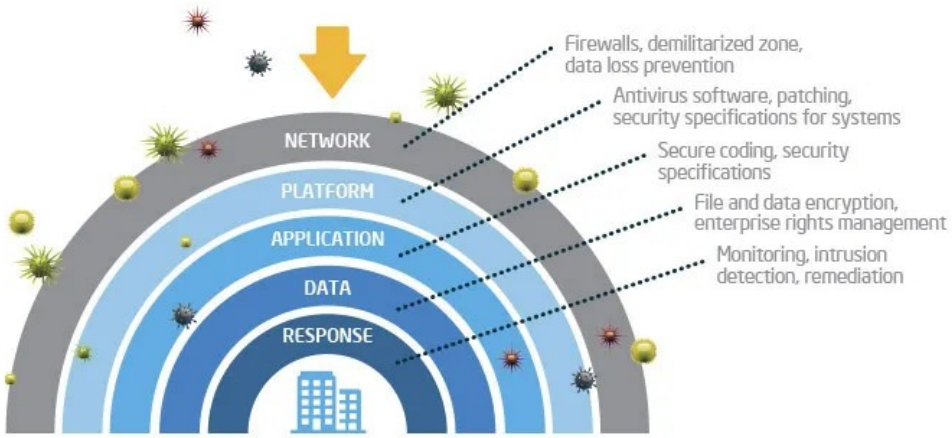
Portanto, é evidente que as pequenas empresas devem estar atentas a essas ameaças, implementando medidas de segurança para proteger seus sistemas. Como Mohseni (2022) enfatiza, a segurança cibernética é uma responsabilidade contínua, especialmente para as pequenas empresas, que muitas vezes não possuem os recursos de grandes corporações para proteger adequadamente suas infraestruturas digitais. Diante das vulnerabilidades comuns observadas em pequenas empresas, torna-se necessário discutir estratégias e soluções para mitigar esses riscos e fortalecer sua postura de segurança.

2.2.4 Estratégias para Mitigação de Riscos

A proteção de redes de pequenas empresas envolve uma série de medidas, e uma das mais importantes é a implementação de *firewalls* adequados. A escolha do *firewall* certo para o tamanho e escopo de uma rede empresarial é importante para

garantir sua segurança. De acordo com Mohseni (2022), a função principal de um *firewall* de rede é bloquear portas que os invasores costumam utilizar para acessar uma organização e roubar informações. Essa barreira ajuda a prevenir o acesso não autorizado, protegendo os dados e garantindo a integridade da rede. No entanto, para uma proteção mais eficiente, o firewall deve ser compreendido como parte de uma estratégia de defesa em profundidade, que abrange desde a infraestrutura de rede até a resposta a incidentes, conforme ilustrado na Figura 4.

Figura 4 - Camadas de Segurança



Fonte: DataDrivenInvestor, 2019

Além disso, em um contexto de crescente ameaça, os *firewalls* desempenham um papel vital na mitigação de ataques que buscam interromper a operação das redes. Ataques como Negação de Serviço (DoS) e Negação de Serviço Distribuída (DdoS) podem prejudicar seriamente a funcionalidade dos negócios, sobrecarregando os recursos da rede e afetando a confiança dos clientes (SAHA & ANWAR, 2023). A sobrecarga ocorre quando um grande volume de solicitações maliciosas consome a largura de banda, a capacidade de processamento ou a memória dos servidores, tornando os serviços indisponíveis para usuários legítimos. Assim, um *firewall* bem configurado não apenas impede o acesso não autorizado, mas também ajuda a mitigar os impactos de ataques cibernéticos, garantindo que a rede continue funcionando mesmo diante de tentativas de sobrecarga.

Apesar de sua importância, o uso de *firewalls* em pequenas empresas ainda é limitado. Segundo Pawar e Palivela (2022), "*firewalls* são usados por 10,2% das

PMEs", o que indica uma lacuna significativa na implementação de medidas de segurança. Esse baixo índice pode ser atribuído, em parte, à falta de conhecimento e conscientização, conforme apontado por Chidukwani et al. (2022), que destacam falhas na implementação de *firewalls,* mesmo quando esses recursos estão disponíveis nos sistemas operacionais. A deficiência na adoção de *firewalls* é uma falha crítica, considerando que esses dispositivos são a linha de defesa primária contra diversas ameaças externas, conforme mostrado na figura 4.

Outro aspecto importante é que, além da segurança na camada de rede, as pequenas empresas devem considerar a proteção das camadas de aplicativos e dados. Pawar e Palivela (2022) enfatizam a importância de focar na segurança de aplicativos voltados ao público, como *Application Programming Interfaces* (APIs) e portais *web*, antes de avançar para camadas internas da rede. Este enfoque nas camadas superiores da segurança ajuda a garantir que, mesmo com um *firewall* eficiente, as informações sensíveis não sejam expostas a riscos adicionais.

Em resumo, o uso de firewalls adequados é um passo decisivo para proteger as redes de pequenas empresas contra ataques externos. No entanto, além da instalação do *firewall*, é necessário que as empresas invistam em treinamento e conscientização sobre a sua configuração e manutenção, criando uma abordagem de segurança em camadas que leve em consideração todas as vulnerabilidades possíveis. A implementação de *firewalls*, junto com medidas de proteção em outras camadas, pode ser determinante para reduzir os riscos de ataques cibernéticos.

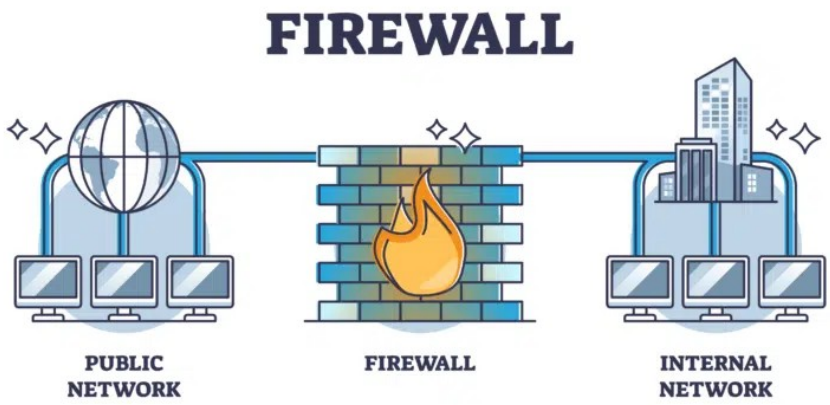
2.3 Firewalls como Mecanismo de Defesa

A segurança das redes modernas depende de uma série de mecanismos integrados, com os *firewalls* destacando-se como barreiras essenciais para prevenir e detectar acessos não autorizados. Eles constituem o primeiro nível de defesa na proteção dos sistemas de informação, atuando na filtragem e controle do tráfego entre ambientes internos e externos.

2.3.1 Função e Necessidade de Firewalls

Os *firewalls* são dispositivos fundamentais para a segurança de redes, pois estabelecem uma barreira que permite apenas a passagem de tráfego autorizado, contribuindo para a prevenção de ataques cibernéticos. Conforme Enoka (2023), todo tráfego, seja de acesso a um site, envio de mensagens ou e-mails, deve passar por pelo menos um *firewall*. A estratégia de prevenir ataques ou detectá-los rapidamente é crucial para proteger os ativos de informação, e os *firewalls* são o principal método de prevenção (COLE, 2009; ENOKA, 2022). Eles permitem o acesso controlado ao mundo externo, conforme mostrado na figura 5, impondo regras que definem quais tipos de tráfego podem entrar ou sair de uma rede, tornando-os adequados na contenção de ameaças (STALLINGS, 2011).

Figura 5 - Firewall



Fonte: EXA, 2022

Historicamente, a evolução da segurança de redes passou de Listas de Controle de Acesso (ACLs), que se mostraram insuficientes, para a implementação de *firewalls,* que oferecem um controle mais refinado e adaptado às necessidades atuais (Alsaqour et al., 2021). Um *firewall* pode ser composto por *hardware*, *software* ou uma combinação de ambos, configurado para analisar o tráfego e bloquear acessos de fontes não confiáveis (SINGH & KAUR, 2024). Essa abordagem não só reforça a segurança interna, mas também possibilita o registro e a auditoria de eventos suspeitos, contribuindo para uma resposta rápida a incidentes.

A eficácia dos *firewalls* depende essencialmente de um projeto e configuração corretos, pois um *firewall* mal configurado pode se tornar uma vulnerabilidade, permitindo que ataques contornem as barreiras de proteção. Essa limitação ressalta a importância de integrá-los em uma estratégia de segurança em camadas, onde outros mecanismos como sistemas de detecção de intrusão, antivírus e políticas de acesso rigorosas atuam em conjunto. Assim, os *firewalls* são uma peça chave em um sistema de defesa mais amplo, que deve ser continuamente atualizado e monitorado. Em síntese, eles são indispensáveis na arquitetura de segurança das redes, por sua capacidade de controlar o fluxo de informações e reduzir a exposição a ameaças externas. A evolução dos mecanismos de proteção evidencia a necessidade contínua de aprimorar as estratégias de segurança, sendo a correta implementação e manutenção dos *firewalls*, associada a uma abordagem em camadas, fundamental para garantir a integridade e a confiabilidade dos sistemas.

Firewalls são indispensáveis para a proteção de redes, principalmente ao compensar deficiências em mecanismos de autenticação e na segurança de aplicações. Em muitos ambientes, serviços e aplicativos não implementam, por padrão, protocolos robustos de autenticação e criptografia devido à complexidade e aos custos associados. Essa fragilidade permite que usuários mal-intencionados se infiltrem nas redes, pois os sistemas não conseguem diferenciar usuários legítimos de agentes fraudulentos (COLE, 2009). A utilização de softwares gratuitos ou de baixo custo, frequentemente empregados para acesso remoto, transferência de arquivos e gerenciamento de e-mail, pode introduzir vulnerabilidades por não serem otimizados com os mais elevados padrões de segurança.

Diante desse cenário, o *firewall* atua como uma camada adicional de defesa, realizando a inspeção, o registro e o controle do tráfego de rede. Essa função é crucial para identificar e bloquear tentativas de intrusão, como o *spoofing* (falsificação de endereços) e ataques automatizados por *scanners* e *crackers* (COLE, 2009). Assim, o *firewall* não só impede que vulnerabilidades existentes sejam exploradas, mas também contribui para a manutenção da integridade e confidencialidade dos dados trafegados. O uso de *firewalls* se justifica como uma resposta necessária às fragilidades presentes tanto na autenticação dos sistemas quanto na segurança dos *softwares* utilizados, consolidando-se como um componente importante na arquitetura de segurança de qualquer organização.

2.3.2 Características do Firewall

Os *firewalls* representam elementos essenciais na segurança de redes, constitindo o primeiro nível de defesa na proteção dos sistemas de informação, atuando como barreiras estratégicas que controlam o fluxo de tráfego entre ambientes internos e externos, protegendo a integridade dos sistemas. Para que essa proteção seja efetiva, os *firewalls* devem ser configurados para que todo o tráfego, tanto de saída quanto de entrada, passe por um ponto único de controle. Essa abordagem permite a aplicação rigorosa das políticas de segurança definidas, autorizando apenas os acessos que se enquadram nos critérios preestabelecidos (STALLINGS, 2011).

Conforme Stallings (2011), essa estratégia visa bloquear acessos diretos à rede, obrigando que todas as comunicações sejam filtradas. Ele também destaca que a adoção de técnicas diversificadas, como controle de serviço, direção, usuário e comportamento, aprimora a capacidade do *firewall* em gerenciar o tráfego, identificando e mitigando potenciais atividades suspeitas. Essa centralização do controle traz vantagens significativas, como a simplificação da gestão de segurança e a facilitação da auditoria e monitoramento de eventos críticos. Ao concentrar funções de filtragem, tradução de endereços e até o suporte a *Virtual Private Networks* (VPNs) em uma única plataforma, o *firewall* previne acessos não autorizados e contribui para a detecção precoce de tentativas de intrusão.

No entanto, é importante econhecer que, apesar de sua confiabilidade, os *firewalls* possuem limitações inerentes, eles podem ser contornados por ataques que exploram vulnerabilidades internas ou métodos alternativos de conexão, como dispositivos móveis ou redes sem fio mal configuradas. Dessa forma, a proteção oferecida pelo *firewall* deve ser encarada como parte de uma estratégia de segurança mais ampla e integrada. Sendo assim, a eficácia dos *firewalls* depende não apenas de sua correta configuração e capacidade de filtrar o tráfego, mas também da integração com outras medidas de segurança que mitigam vulnerabilidades internas e ataques indiretos. Assim, embora indispensáveis na proteção das redes, os *firewalls* devem ser complementados por práticas de segurança abrangentes, garantindo uma defesa sólida e resiliente contra ameaças variadas. Para atender a essas características e necessidades diversas de proteção,

os *firewalls* são desenvolvidos em diferentes arquiteturas e com variadas funcionalidades, dando origem a distintos tipos.

2.3.3 Tipos de Firewalls

Para compreender a funcionalidade e aplicabilidade dos *firewalls*, é necessário analisar os diferentes tipos existentes, suas características e casos de uso. Os *firewalls* podem ser classificados, de maneira geral, em duas categorias principais: *firewalls* de *hardware* e *firewalls* de *software.*

O *firewall* de *hardware* é um dispositivo físico posicionado na infraestrutura de rede para controlar o tráfego antes que ele atinja os dispositivos internos. Já o firewall de *software* é um programa instalado diretamente em um *endpoint*, como um computador ou servidor, exigindo configurações adicionais para um funcionamento satisfatório. Ambas as abordagens têm o objetivo de reduzir a superfície de ataque, ou seja, os pontos vulneráveis passíveis de exploração por agentes mal-intencionados (ENOKA, 2022).

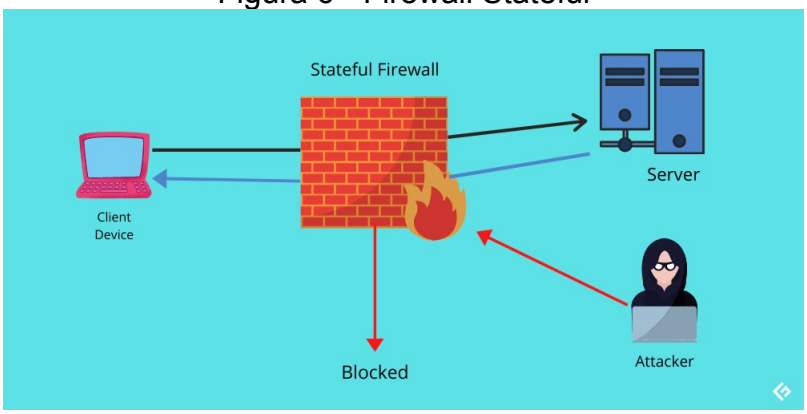
Outra classificação relevante é baseada na localização e na forma como o *firewall* opera dentro da rede. O *firewall* de perímetro, por exemplo, está posicionado na fronteira entre a rede privada e redes externas, como a internet. Ele pode ser implementado tanto em *hardware* quanto em software e funciona como a primeira camada de defesa contra tráfego potencialmente malicioso. Essa abordagem garante que apenas pacotes de dados permitidos pelas regras estabelecidas possam transitar entre redes internas e externas (ENOKA, 2022).

Além da distinção entre *hardware* e *software*, os *firewalls* podem ser diferenciados pelo método de análise do tráfego. O *firewall* de filtragem de pacotes verifica individualmente cada pacote de dados que entra ou sai da rede, comparando suas informações com um conjunto de regras predefinidas. Caso o pacote atenda aos critérios estabelecidos, ele será permitido ou bloqueado conforme a configuração aplicada.

Os *firewalls* também podem ser categorizados como *stateful* (com estado) e *stateless* (sem estado). O *firewall stateful,* conforme ilustrado na figura 6, mantém um registro das conexões ativas e monitora cada fluxo de dados como parte de uma comunicação contínua, permitindo um controle mais refinado do tráfego. Esse

método possibilita uma resposta mais precisa às ameaças, pois considera o contexto das conexões.

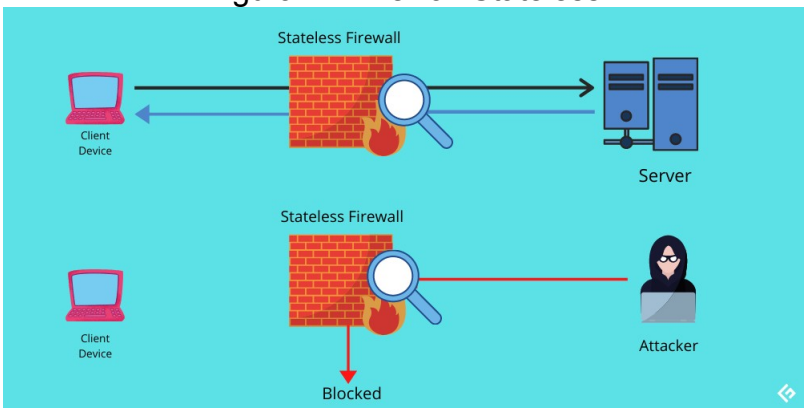
Figura 6 - Firewall Stateful



Fonte:GeekFlare, 2024

Em contrapartida, o *firewall stateless,* conforme ilustrado na figura 7, trata cada pacote de forma independente, sem armazenar informações sobre conexões anteriores, o que pode resultar em menor eficácia no bloqueio de tráfego malicioso baseado em padrões de comportamento (ENOKA, 2022).

Figura 7 - Firewall Stateless



Fonte: GeekFlare, 2024

Dentre os exemplos de *firewalls stateful* amplamente utilizados, destacam-se o *iptables*, presente em sistemas Linux, e o *pfSense*, uma solução baseada em *software* de código aberto. A maioria dos sistemas operacionais modernos, como Windows e macOS, já inclui um *firewall* baseado em *host,* projetado para filtrar o tráfego em nível individual de dispositivo. Embora esses *firewalls* sejam pré-

configurados para fornecer segurança básica, em ambientes Linux, geralmente, é necessária uma configuração manual para obter um nível de proteção adequado.

Além da classificação baseada em estado, é possível distinguir os *firewalls* pela profundidade da inspeção. *Firewalls* tradicionais (*Packet Filtering*) operam nas Camadas 3 e 4 do modelo OSI, tomando decisões baseadas apenas nos cabeçalhos dos pacotes (IP de origem/destino e portas TCP/UDP). Eles agem como porteiros que verificam apenas o envelope de uma carta, sem ler o conteúdo. Por outro lado, os *Firewalls* de Próxima Geração (NGFW) realizam a Inspeção Profunda de Pacotes (DPI), atuando até a Camada 7 (Aplicação). Isso permite analisar o *payload* (conteúdo) do pacote, identificando assinaturas de malware, tentativas de injeção de código, como SQL *Injection*, ou uso indevido de protocolos, mesmo que a conexão ocorra em uma porta autorizada, como a 80 ou 443.

Considerando as ameaças cibernéticas atuais, a adoção de múltiplas camadas de defesa é recomendada. A combinação de um *firewall* de perímetro com um *firewall* baseado em *host* proporciona uma estratégia mais reforçada, mitigando riscos de acessos não autorizados e fortalecendo a segurança da rede como um todo. Portanto, a escolha do tipo de *firewall* deve estar alinhada às necessidades específicas do ambiente protegido, assegurando que o ambiente mantenha um nível adequado de proteção sem comprometer sua operação.

2.3.4 Vulnerabilidades dos Firewalls

Embora os *firewalls* constituam a primeira linha de defesa na segurança de redes, eles apresentam vulnerabilidades que podem comprometer a proteção completa de uma rede (ALSAQUOR, MOTMI & ABDELHAQ 2021). Os autores destacam cinco principais fragilidades: ataques internos, gestão inadequada de *patches*, erros de configuração, falta de inspeção profunda de pacotes (DPI) e vulnerabilidade a ataques DDoS. Cada uma dessas lacunas expõe riscos específicos. Por exemplo, ataques internos desafiam a noção tradicional de que *firewalls* externos são suficientes para proteger a rede, já que agentes maliciosos podem operar dentro do perímetro organizacional. Além disso, a dependência excessiva de *firewalls* básicos, sem funcionalidades avançadas como DPI, limita a capacidade de detectar ameaças ocultas em pacotes de dados (ALSAQUOR, MOTMI & ABDELHAQ 2021).

A argumentação central reside na necessidade de uma abordagem multifacetada para a segurança. Por um lado, ataques internos evidenciam que a segmentação de redes com *firewalls* internos é determinante para retardar invasores. No entanto, isso não elimina riscos, pois a falta de atualizações regulares de *patches* de segurança, muitas vezes negligenciadas por equipes sobrecarregadas, mantém portas abertas para exploração. Um *firewall* desatualizado é tão perigoso quanto sua ausência, já que vulnerabilidades conhecidas podem ser facilmente aproveitadas por cibercriminosos.

Outro ponto crítico são os erros de configuração, que transformam o *firewall* de um aliado em um obstáculo frágil. Configurações equivocadas, como regras excessivamente permissivas ou falhas na definição de políticas, não só desperdiçam recursos, mas também criam brechas exploráveis. Paralelamente, a ausência de DPI em *firewalls* convencionais permite que ataques de *spoofing* e *malware* contornem barreiras superficiais, já que a análise limita-se a metadados, como origem e destino dos pacotes.

Quanto aos ataques DDoS, a discussão revela uma contradição: embora *firewalls* auxiliem na mitigação, sua arquitetura pode ser sobrecarregada por tráfego malicioso massivo, especialmente em ataques baseados em protocolo. Soluções complementares, como serviços de *scrubbing*, são necessárias para filtrar tráfego antes que alcance a rede, reduzindo a pressão sobre os *firewalls* (ALSAQUOR, MOTMI & ABDELHAQ 2021).

Portanto, as vulnerabilidades dos *firewalls* não invalidam sua utilidade, mas exigem que sejam integrados a outras camadas de segurança. A gestão proativa de *patches*, a configuração rigorosa, a adoção de tecnologias como DPI e a combinação com serviços anti-DDoS são medidas indispensáveis para compensar suas limitações. A cibersegurança eficiente depende menos de ferramentas isoladas e mais de uma estratégia holística, que reconheça e contorne as fragilidades inerentes a cada componente (ALSAQUOR, MOTMI & ABDELHAQ 2021). Apesar das vulnerabilidades inerentes, os *firewalls* continuam sendo uma ferramenta essencial na segurança de redes, apresentando um conjunto de vantagens e desvantagens que devem ser avaliadas para sua implementação eficaz.

2.3.5 Vantagens e Desvantagens de Firewalls

Os *firewalls* desempenham um papel essencial na segurança de redes ao monitorar e controlar o tráfego de dados. No entanto, como qualquer tecnologia de segurança, eles possuem tanto vantagens quanto desvantagens, que devem ser consideradas antes de sua implementação.

Uma das principais vantagens dos *firewalls* é a capacidade de monitoramento de tráfego. Eles analisam os pacotes de dados que transitam pela rede, determinando se devem ser bloqueados ou permitidos. Isso garante um controle rigoroso sobre a entrada e saída de informações, reduzindo o risco de ameaças virtuais (ALSAQUOR, MOTMI & ABDELHAQ 2021). Outra vantagem significativa é a prevenção contra ameaças como *trojans*, *keyloggers* e ataques de *hackers*. *Firewalls* bloqueiam códigos maliciosos antes que possam se infiltrar na rede e comprometer sistemas, proporcionando uma camada de proteção contra *malwares* disfarçados. Esse aspecto é crucial para manter a integridade dos sistemas empresariais e individuais. Ademais, os *firewalls* aumentam a segurança ao impedir acessos não autorizados, dificultando a atuação de *hackers*. Sem um *firewall*, redes e dispositivos tornam-se vulneráveis a invasões, o que pode resultar no roubo de informações confidenciais ou na disseminação de códigos maliciosos (ALSAQUOR, MOTMI & ABDELHAQ 2021).

Por outro lado, a implementação de firewalls impõe desafios técnicos, gerando um impacto na performance da infraestrutura. Um dos principais problemas é a redução do desempenho dos sistemas. A verificação constante de pacotes consome recursos computacionais, podendo impactar negativamente a velocidade e a eficiência dos dispositivos e da rede (ALSAQUOR, MOTMI & ABDELHAQ 2021). Esse impacto é mais perceptível em *firewalls* baseados em *software*, que dependem do *hardware* do computador para seu funcionamento.

Outra desvantagem relevante é a possibilidade de restrição a usuários legítimos. Políticas excessivamente rigorosas podem impedir que funcionários acessem determinados recursos necessários para suas atividades, reduzindo a produtividade. Dessa forma, é essencial que as regras de *firewall* sejam bem configuradas para equilibrar segurança e usabilidade. Além disso, *firewalls* podem criar uma falsa sensação de segurança. Muitos usuários podem acreditar que, ao contar com um *firewall*, estão completamente protegidos contra ameaças digitais,

negligenciando outras boas práticas de segurança, como atualização de *software* e uso de antivírus.

Outro aspecto a ser considerado é o custo total de propriedade (aquisição e licenciamento). *Firewall*s d*e hardwa*re, por exemplo, frequentemente exigem investimentos elevados, tornando-se barreiras financeiras para pequenas empresas (ALSAQUOR, MOTMI & ABDELHAQ 2021). Assim, é necessário avaliar o custo-benefício dessa solução conforme as necessidades específicas de cada organização. No contexto deste trabalho, foi evidenciado que o Raspberry Pi apresenta um custo de implementação inferior. Por fim, o*s firewall*s possuem limitações no que diz respeito à prevenção de ataques internos. Como são projetados para barrar ameaças externas, eles não conseguem impedir que indivíduos já autenticados na rede realizem atividades maliciosas.

Os *firewalls* são ferramentas indispensáveis para a segurança cibernética, oferecendo proteção contra diversas ameaças ao monitorar e controlar o tráfego de rede. Entretanto, sua implementação deve ser bem planejada para evitar impactos negativos, como restrição indevida de usuários legítimos, queda de desempenho e altos custos. Além disso, é indispensável que seu uso seja complementado por outras estratégias de segurança, garantindo uma proteção abrangente contra ameaças digitais. Para aproveitar os benefícios dos *firewalls* e reduzir suas limitações, é importante adotar boas práticas na implementação e na gestão desses sistemas, assegurando que funcionem como uma barreira de defesa consistente.

2.3.6 Boas Práticas de Implementação e Gestão

A implementação de *firewalls* é um dos pilares fundamentais para a segurança de redes, sendo essencial para a proteção contra acessos não autorizados e ataques cibernéticos. No entanto, para garantir sua eficácia, é necessário seguir diretrizes e boas práticas que envolvem desde a segurança do próprio *firewall* até a definição precisa de zonas de segurança e listas de controle de acesso (ACLs) (ALSAQUOR, MOTMI & ABDELHAQ 2021).

Um dos aspectos mais críticos na implementação de um *firewall* é a sua própria segurança. Um *firewall* desatualizado ou mal configurado pode se tornar um ponto de vulnerabilidade em vez de um mecanismo de proteção competente. Assim, é recomendável que os administradores garantam a atualização constante do

firmware, a remoção de contas padrão e a utilização de credenciais resistentes. Ademais, o uso de protocolos de gerenciamento seguros, como o *Simple Network Management Protocol* (SNMP) configurado com *strings* de comunidade seguras, minimiza o risco de exploração por atacantes (ALSAQUOR, MOTMI & ABDELHAQ 2021).

Outra prática essencial é a definição de zonas de segurança dentro da rede. A segmentação adequada dos ativos permite um controle mais refinado sobre os fluxos de tráfego e a aplicação de políticas de segurança adequadas a cada contexto. Servidores que necessitam de acesso público, como os que oferecem serviços de VPN, devem estar em zonas fortemente protegidas, enquanto os dispositivos internos devem permanecer isolados, utilizando endereçamento privado com *Network Address Translation* (NAT) para comunicação externa quando necessário. A estruturação adequada dessas zonas reduz significativamente os vetores de ataque e aprimora a segurança da organização (ALSAQUOR, MOTMI & ABDELHAQ 2021).

A criação e gestão de listas de controle de acesso (ACLs) também desempenham um papel vital na proteção da rede. ACLs devem ser configuradas para filtrar o tráfego entre as zonas, especificando endereços *Internet Protocol* (IP) de origem e destino, além de definir portas e protocolos permitidos. Uma prática recomendada é a implementação de uma regra final de "negar tudo" para bloquear acessos não explicitamente autorizados. Adicionalmente, é aconselhável desativar interfaces de administração expostas e evitar o uso de protocolos não criptografados, mitigando assim ameaças externas (ALSAQUOR, MOTMI & ABDELHAQ 2021).

Além da configuração do *firewall*, a gestão de *logs* e serviços complementares é essencial para o monitoramento e a resposta a incidentes. Caso o *firewall* possua funcionalidades adicionais, como servidor *Dynamic Host Configuration Protocol* (DHCP) ou sistema de prevenção de intrusão (IPS), é necessário configurá-las de maneira adequada. A gravação de *logs* em servidores dedicados é uma prática recomendada, permitindo auditoria e análise posterior de eventos suspeitos (ALSAQUOR, MOTMI & ABDELHAQ 2021).

Por fim, antes da implementação definitiva, é imprescindível testar a configuração do *firewall* em um ambiente controlado. Testes de vulnerabilidade e *pentests* devem ser conduzidos para garantir que as regras definidas estejam

alinhadas aos objetivos de segurança da organização. Além disso, a manutenção de *backups* das configurações do *firewall* assegura uma recuperação rápida em caso de falha ou comprometimento (ALSAQUOR, MOTMI & ABDELHAQ 2021).

Dessa forma, a correta implementação de um *firewall* requer uma abordagem estruturada, baseada na adoção de boas práticas que englobam desde a segurança do próprio *firewall* até a segmentação de redes, configuração de ACLs e monitoramento contínuo. Com essas medidas, é possível estabelecer uma proteção para os ativos da organização, reduzindo os riscos de ameaças cibernéticas.

2.4 Uso do Raspberry Pi como Firewall

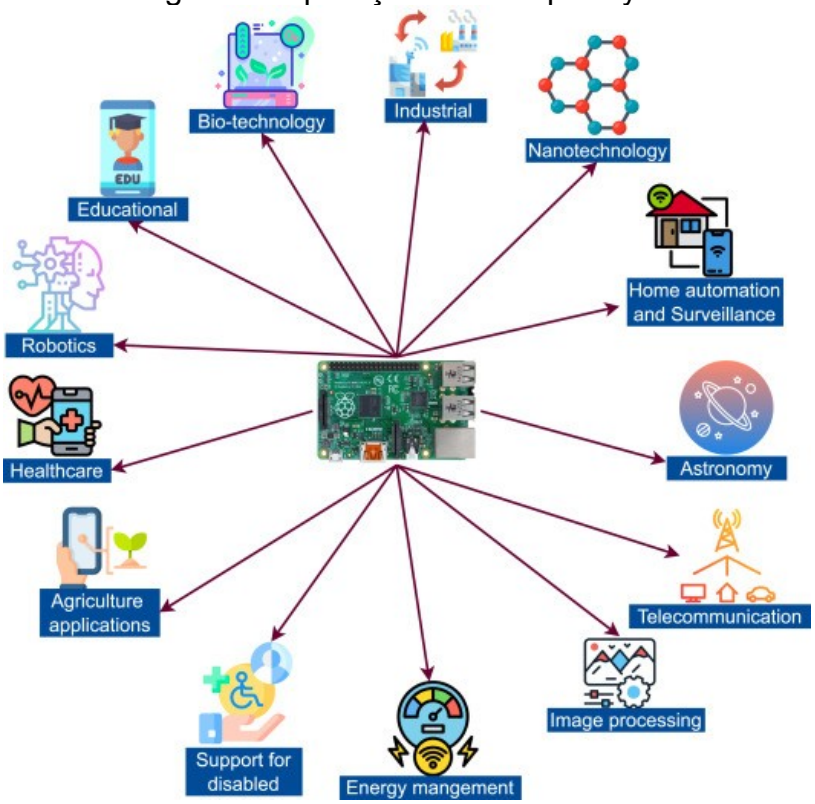
O Raspberry Pi é um computador de placa única amplamente adotado em diversas áreas, destacando-se por seu baixo custo comparado a dispositivos de segurança dedicados, consumo reduzido de energia e tamanho compacto. Tipicamente, esses dispositivos operam com distribuições Linux otimizadas, como o Raspberry Pi *Operating System* (OS), o que permite grande flexibilidade na configuração de softwares de segurança. Essas características tornam essa plataforma uma solução viável para aplicações que exigem flexibilidade e eficiência, desde automação residencial até sistemas de segurança e redes de comunicação, conforme mostrado na figura 8.

No contexto das telecomunicações, Mathe et al. (2024) destacam que o Raspberry Pi pode ser empregado na implementação de servidores de Voz sobre Protocolo de Internet (VoIP), sistemas *Private Branch Exchange* (*PABX*) e *gateways* de Protocolo de Iniciação de Sessão (SIP), proporcionando alternativas acessíveis para infraestrutura de comunicação. Além disso, sua capacidade de processar e gerenciar tráfego de rede faz com que seja uma ferramenta útil para atividades como monitoramento de redes, análise de pacotes e controle de acessos.

No campo da segurança da informação, o Raspberry Pi pode ser empregado como uma solução para administração de *firewalls*, permitindo a implementação de políticas de filtragem de tráfego e prevenção contra ataques cibernéticos. Mathe et al. (2024) ressaltam sua utilização como analisador de pacotes, monitoramento de redes e *firewalls,* permitindo a identificação de padrões suspeitos e potenciais ameaças. Essa funcionalidade é essencial para ambientes que demandam soluções de segurança acessíveis e altamente personalizáveis. A versatilidade do Raspberry

Pi o torna uma alternativa viável para pequenas e médias infraestruturas que necessitam de ferramentas de segurança e gerenciamento de redes de baixo custo.

Figura 8 - Aplicações do Raspberry Pi



Fonte: MATHE, 2024

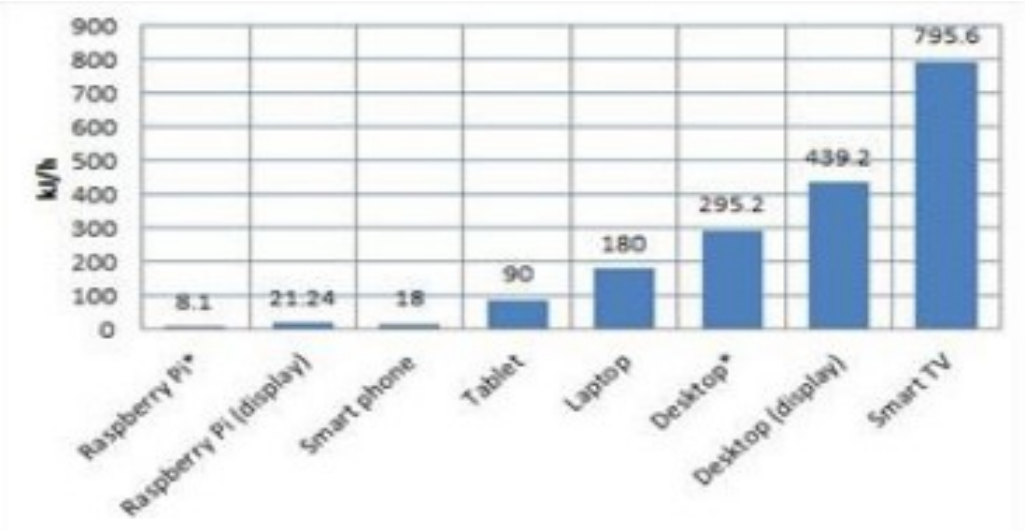
A viabilidade do Raspberry Pi como *firewall* é reforçada por suas próprias características, especialmente sua eficiência energética e o baixo consumo de recursos.

2.4.1 Eficiência Energética e Consumo de Recursos

A crescente preocupação com a eficiência energética e a sustentabilidade tem impulsionado a busca por dispositivos com baixo consumo de energia, especialmente em ambientes de uso contínuo. Nesse contexto, o Raspberry Pi se destaca como uma solução energeticamente eficiente em comparação a outros dispositivos de computação mais tradicionais. De acordo com Akshay et al. (2018), o Raspberry Pi opera com apenas 2,25 watts de consumo energético em seu uso padrão, o que resulta em uma economia significativa em comparação a outros dispositivos como *notebooks*, *desktops* e televisores.

Em termos de consumo de energia, o Raspberry Pi é muito mais eficiente do que equipamentos convencionais. Por exemplo, enquanto um *notebook* consome cerca de 50 watts (equivalente a 180 kJ/h), o Raspberry Pi utiliza apenas 2,25 watts (resultando em 8,1 kJ/h). Isso significa que o Raspberry Pi consome apenas 4,5% da energia utilizada por um *notebook*, resultando em uma redução de aproximadamente 95,5% no consumo energético (AKSHAY et al., 2018). Comparando com outros dispositivos de uso comum, como um *desktop*, que consome 82 watts e atinge 295,2 kJ/h sem o monitor, o Raspberry Pi se mostra uma opção muito mais eficiente para tarefas que não demandam grandes recursos computacionais, conforme mostrado na figura 9.

Figura 9 - Eficiência Energética do Raspberry Pi



Fonte: AKSHAY, 2018

Além da eficiência energética, a portabilidade do Raspberry Pi também é um ponto importante. Ao contrário de *desktops* e *smart TVs*, que são pesados e não portáteis, o Raspberry Pi tem o tamanho de um cartão de crédito, facilitando seu transporte e utilização em diversos cenários. Em termos de consumo de energia, a flexibilidade de conexão a diferentes tipos de *displays*, como monitores *Liquid Crystal Display* (LCD) e televisores, permite que o Raspberry Pi se adapte a uma variedade de usos com baixo impacto energético. A comparação com dispositivos como *smartphones* e *tablets* também demonstra que, apesar da mobilidade superior dos *smartphones*, o Raspberry Pi oferece um equilíbrio entre portabilidade e

capacidade de processamento, mantendo alto nível de eficiência energética mesmo em aplicações contínuas.

A utilização em larga escala do Raspberry Pi pode ter um impacto expressivo na redução do consumo de energia. Em um exemplo prático, foi demonstrado que, em uma escola com 400 *desktops* funcionando 12 horas por dia, o consumo semanal seria de aproximadamente 1.968 kWh. Contudo, ao substituir esses equipamentos por Raspberry Pi, o consumo semanal cairia para cerca de 54 kWh, resultando em uma redução de 97,4% no gasto energético (AKSHAY et al., 2018). Essa diferença não apenas diminui os custos operacionais, mas também contribui para a mitigação de impactos ambientais, ao reduzir significativamente a demanda por energia elétrica.

Em resumo, o consumo reduzido de energia do Raspberry Pi mostra que ele pode mudar a maneira como os recursos energéticos são utilizados, principalmente em aplicações que não demandam alto poder de processamento. Seu baixo gasto energético, aliado à portabilidade, torna o dispositivo uma opção atraente para ambientes educacionais e projetos de baixo custo, destacando-se como uma solução sustentável na computação atual. O uso de equipamentos como o Raspberry Pi contribui para uma atuação mais econômica e com menor impacto ambiental.

2.4.2 Impacto do Hardware Limitado

A implementação de *firewalls* em dispositivos com recursos limitados, como o Raspberry Pi, levanta questões cruciais sobre o equilíbrio entre segurança e desempenho. A capacidade de processamento e memória de um dispositivo embarcado pode afetar diretamente a eficácia de um *firewall*, especialmente quando ele precisa realizar tarefas intensivas, como a filtragem de pacotes e a análise de tráfego de rede. Rumelioglu (2005) aponta que, embora os *firewalls* sejam eficazes em prevenir o acesso não autorizado à rede, eles enfrentam desafios consideráveis quando se trata de impedir ataques originados dentro da própria rede.

Durante o processo de filtragem, o *firewall* consome recursos significativos do sistema. Esse consumo é intensificado em dispositivos com hardware limitado, como o Raspberry Pi, que possui uma capacidade de processamento e memória inferior a outros servidores mais robustos. Em tais sistemas, o *firewall* pode consumir uma

parte considerável dos recursos, tornando-se, paradoxalmente, o próprio gargalo da rede. Isso ocorre especialmente quando o tráfego de rede é alto ou quando as regras de filtragem são complexas. A falta de recursos computacionais suficientes pode resultar em um desempenho comprometido, onde o *firewall* não consegue lidar adequadamente com o tráfego, afetando a performance global da rede (RUMELIOGLU, 2005).

Além disso, a implementação de diversos *firewalls* distribuídos em diferentes segmentos da rede, com o objetivo de evitar que um único equipamento concentre todo o processamento do tráfego e se torne um ponto de estrangulamento do desempenho, pode ser uma alternativa viável. No entanto, essa estratégia introduz um novo desafio operacional: a necessidade de gerenciar e sincronizar regras, políticas e registros entre vários dispositivos. Essa coordenação aumenta a complexidade administrativa e demanda maior capacidade de processamento do equipamento responsável pelo controle central. Em cenários que utilizam hardware limitado, essa comunicação constante entre os *firewalls* pode resultar em uso excessivo de recursos, dificultando a escalabilidade e comprometendo a eficiência do sistema como um todo.

Rumelioglu (2005) também observa que, em sistemas de *firewall* baseados em *host*, a instalação de *firewalls* em cada dispositivo da rede pode resultar em um impacto negativo no desempenho geral do sistema. Isso ocorre porque cada *host* precisa alocar recursos para o processamento do tráfego e a manutenção de suas regras de segurança. Em dispositivos com recursos limitados, como o Raspberry Pi, o impacto no desempenho pode ser mais acentuado, resultando em uma diminuição da eficiência do sistema e aumento da latência na rede.

Em comparação, os *firewalls* baseados em rede, que funcionam no *gateway* da rede, tendem a minimizar o impacto no desempenho dos *hosts* individuais. Isso ocorre porque o processamento do tráfego é centralizado, reduzindo a carga nos dispositivos finais. No entanto, como observado por Rumelioglu (2005), essa abordagem também apresenta limitações, como a possibilidade de perder pacotes se o tráfego de rede for muito intenso, além de não impedir ataques que ocorram dentro da rede e que não atravessam o *gateway*.

Em resumo, o impacto do hardware limitado no desempenho de *firewalls* é um fator crítico que deve ser considerado ao projetar sistemas de segurança em dispositivos embarcados, como o Raspberry Pi. A implementação de *firewalls* requer

um equilíbrio cuidadoso entre a capacidade de processamento, as necessidades de segurança e a eficiência do sistema. Em dispositivos com recursos restritos, a sobrecarga do *firewall* pode comprometer o desempenho da rede, tornando essencial a escolha de uma abordagem que minimize esse impacto enquanto ainda oferece proteção eficaz. No contexto do Raspberry Pi, que utiliza um sistema operacional baseado em Linux, a configuração de *firewalls* como o nftables ou *Uncomplicated Firewall* (UFW) é comum, mas o desempenho pode ser diretamente impactado pelas especificações do hardware e pela otimização do software.

2.5 Desempenho e Estabilidade de Firewalls de Baixo Custo

A adoção de *firewalls* de baixo custo, como o Raspberry Pi, apresenta-se como uma opção interessante para pequenas empresas e usuários domésticos. Contudo, seu desempenho e sua estabilidade podem ser limitados quando comparados a soluções mais avançadas. Nesse cenário, torna-se importante entender quais fatores influenciam a eficiência e a confiabilidade desses dispositivos.

2.5.1 Problemas de Desempenho e Impacto na Rede

O desempenho de um *firewall* é um fator determinante para a sua efetividade em uma rede. Segundo Konikiewicz e Markowski (2017), os *firewalls* devem garantir o nível adequado de segurança, mas sem comprometer o desempenho da rede, evitando o aumento excessivo da latência nos pacotes. Esse equilíbrio entre segurança e *performance* é relevante, pois *firewalls* que introduzem latência excessiva podem prejudicar a eficiência geral da rede. A largura de banda máxima do firewall e o atraso introduzido são critérios essenciais para avaliar a eficácia de um dispositivo de segurança, especialmente em cenários de alto tráfego.

Além disso, a resistência a ataques de negação de serviço (DoS) é outro critério importante, pois a capacidade de resistir a tais ataques impacta diretamente na estabilidade da rede durante incidentes de segurança. Em ambientes de baixo custo, como o Raspberry Pi, a latência pode ser maior e a capacidade de resistir a esses ataques pode ser limitada, tornando esses dispositivos mais vulneráveis a

falhas sob cargas elevadas. Portanto, a escolha do *firewall* deve considerar não apenas a segurança, mas também o impacto que ele terá no tráfego de dados e na *performance* da rede (KONIKIEWICZ & MARKOWSKI, 2017).

No entanto, a implementação de *firewalls* pode impactar o desempenho da rede. A exigência por *firewalls* levou à sua onipresença, com quase todas as organizações conectadas à Internet utilizando um *firewall*. Segundo Agbenyegah e Asante (2017), "a segurança de rede e desempenho de rede estão inversamente relacionados", com o desempenho da rede sendo negativamente afetado pela implementação de *firewalls*, especialmente quando políticas de segurança mais rígidas são aplicadas. Isso resulta em uma degradação no desempenho da rede, mas, ao mesmo tempo, os *firewalls* desempenham um papel essencial ao proteger o sistema, melhorar a acessibilidade e reduzir o processamento de solicitações ilegítimas, contribuindo para a proteção contra ataques.

2.5.2 Limitações de Hardware e Software

A eficiência de um *firewall* também é fortemente influenciada pelas limitações do hardware e do software utilizados. Konikiewicz e Markowski (2017) destacam que os *firewalls* de *software* "não têm recursos dedicados" e utilizam os recursos do sistema operacional no qual estão instalados, o que os impede de operar de maneira totalmente automatizada. Em soluções de baixo custo, como o Raspberry Pi, o uso de recursos compartilhados pode reduzir significativamente o desempenho, já que o dispositivo não possui a mesma capacidade de processamento ou otimizações dedicadas que *firewalls* de *hardware*, como os modelos Cisco ASA. No contexto do Raspberry Pi, que utiliza um sistema operacional baseado em Linux, a configuração de *firewalls* como o *iptables* ou *Uncomplicated Firewall* (UFW) é comum, mas o desempenho pode ser diretamente impactado pelas especificações do hardware e pela otimização do software.

Por outro lado, os "*firewalls* de *hardware* não dependem de software de terceiros", o que lhes confere maior estabilidade e desempenho dedicado. No entanto, em dispositivos de baixo custo, como o Raspberry Pi, o hardware limitado pode se tornar um gargalo quando comparado a *firewalls* de *hardware* mais robustos, resultando em uma diminuição na eficiência geral. A falta de recursos dedicados em *firewalls* de software exige uma análise cuidadosa sobre como o

hardware do dispositivo pode influenciar o desempenho, especialmente quando se trata de ambientes de alta demanda (KONIKIEWICZ & MARKOWSKI, 2017).

Conforme Bodipudi (2024) discute, os *firewalls* baseados em *host* enfrentam desafios adicionais, como o consumo significativo de recursos do sistema, com a filtragem de tráfego exigindo grande uso de CPU e memória, particularmente em sistemas com capacidade limitada. Esse problema pode ser mitigado por meio da implementação de algoritmos eficientes e conjuntos de regras otimizados, ajustados de acordo com as necessidades específicas do *host*. Nos *firewalls* baseados em rede, por outro lado, o aumento da latência devido à sobrecarga de análise de tráfego, especialmente na inspeção profunda de pacotes, é um problema comum. Bodipudi (2024) sugere que análises de dados e a aplicação de aprendizado de máquina podem ajudar a prever e filtrar tráfego de baixo risco, otimizando os recursos e permitindo que a inspeção profunda seja reservada apenas para tráfego de alto risco. Tais abordagens ajudam a mitigar os desafios de desempenho e aumentar a eficiência dos *firewalls* em ambientes de rede mais complexos.

2.5.3 Efeito das Regras no Desempenho

A forma como as regras são configuradas em um firewall define o nível de eficiência do sistema e seus efeitos no desempenho da rede. De acordo com a análise de Konikiewicz e Markowski (2017), "o *firewall* ideal deve introduzir a menor latência de pacotes na rede e, ao mesmo tempo, fornecer um bom nível de proteção para os dados do usuário". As regras de filtragem de pacotes, embora essenciais para a segurança, podem aumentar o tempo de processamento de cada pacote, o que impacta diretamente na largura de banda e no atraso de transmissão.

O desempenho de *firewalls* de software e hardware é afetado pelo número e complexidade das regras implementadas. *Firewalls* de software, como os baseados em Linux, podem apresentar desempenho semelhante a *firewalls* de *hardware* quando as regras são simples, mas com um número crescente de regras, a *performance* tende a diminuir, especialmente em dispositivos de baixo custo. Segundo Konikiewicz e Markowski (2017), "A largura de banda oferecida pelo *firewall* baseado em PC (*iptables*) foi frequentemente maior do que a oferecida pelo dispositivo de rede dedicado", o que mostra que a escolha das regras e a

configuração do *firewall* podem ter um impacto considerável na *performance* geral, mesmo em sistemas de baixo custo.

2.5.4 Estabilidade e Confiabilidade em Uso Contínuo

A estabilidade e confiabilidade de um *firewall* são características críticas, especialmente em implementações que exigem operação contínua e sem falhas. No caso dos *firewalls* de baixo custo, como o Raspberry Pi, Konikiewicz e Markowski (2017) afirmam que "os *firewalls* de *hardware* e virtuais se mostraram resistentes a ataques de negação de serviço", mostrando que, apesar das limitações de recursos, essas soluções podem manter um nível aceitável de operação sob ataques. No entanto, a "resistência aos ataques DoS" não deve ser vista isoladamente, pois outros fatores, como a capacidade do dispositivo de suportar carga por longos períodos, também afetam sua confiabilidade.

Em ambientes de baixo custo, o hardware limitado pode levar a falhas em situações de uso contínuo, como o superaquecimento do dispositivo ou o esgotamento de recursos de processamento. O desempenho contínuo do *firewall* de software depende não apenas da configuração das regras, mas também da eficiência do sistema operacional e da capacidade do hardware de lidar com o tráfego de dados. Isso implica que, em soluções de baixo custo, é essencial garantir que o dispositivo esteja otimizado para operação contínua, minimizando o impacto de falhas e maximizando a estabilidade a longo prazo.

2.5.5 Riscos de Segurança

Apesar das vantagens de custo e flexibilidade, a utilização de um Raspberry Pi como *firewall* apresenta riscos de segurança próprios da plataforma, os quais exigem atenção e tratamento adequado. Um Raspberry Pi comprometido pode se tornar um ponto de entrada para a rede interna, permitindo que atacantes *bypasssem* as defesas e acessem recursos sensíveis. Os principais riscos incluem:

- Vulnerabilidades do Sistema Operacional: O sistema operacional Linux, embora robusto, pode apresentar vulnerabilidades se não for mantido atualizado com *patches* de segurança. Um Raspberry Pi desatualizado pode ser explorado por atacantes que buscam brechas conhecidas.

- Configuração Inadequada: Erros na configuração das regras do *firewall* (*iptables* ou outras ferramentas) podem criar aberturas na rede. Regras excessivamente permissivas ou mal definidas podem permitir tráfego indesejado, tornando o *firewall* ineficaz.
- *Hardware* Limitado e *Throttling*: Em cenários de alta carga de tráfego ou ataques DDoS, o hardware limitado do Raspberry Pi pode levar a um *throttling* (redução de desempenho devido ao superaquecimento) ou até mesmo a travamentos. Isso pode resultar em uma falha na proteção da rede, deixando-a exposta.
- Falta de Recursos Avançados: *Firewalls* comerciais dedicados geralmente incluem funcionalidades avançadas como Inspeção Profunda de Pacotes (DPI), sistemas de prevenção de intrusão (IPS) integrados e inteligência de ameaças em tempo real. O Raspberry Pi, por si só, pode não oferecer o mesmo nível de proteção sem a implementação e configuração complexa de *softwares* adicionais.
- Ataques Físicos: Por ser um dispositivo pequeno e acessível, o Raspberry Pi pode ser mais suscetível a ataques físicos se não estiver em um ambiente seguro. O acesso físico pode permitir que um atacante obtenha controle total sobre o dispositivo e, consequentemente, sobre a rede.
- Monitoramento e Auditoria Limitados: Sem ferramentas de monitoramento e *logging* bem configuradas, a detecção de atividades suspeitas ou de um comprometimento pode ser mais difícil e demorada, aumentando o tempo de resposta a incidentes.

Para mitigar esses riscos, é recomenda-se a adoção de boas práticas de segurança, como manter o sistema operacional e os *softwares* de *firewall* sempre atualizados, configurar regras de forma rigorosa e testá-las exaustivamente, monitorar constantemente o desempenho e a temperatura do dispositivo, e implementar medidas de segurança física. Ao compreender esses riscos, torna-se possível implementar o Raspberry Pi como *firewall* de maneira mais segura e confiável.

2.6 Métricas de Avaliação de Desempenho

Na avaliação de sistemas de *firewall*, a utilização de métricas adequadas é essencial para medir de forma precisa e objetiva o desempenho da solução. No contexto da análise de *firewalls,* as métricas não se limitam apenas ao desempenho em termos de velocidade, mas também incluem a capacidade de processamento, eficiência no controle de tráfego e a eficácia na filtragem de pacotes, entre outras variáveis. A padronização dessas métricas é abordada por diversos *Request for Comments* (RFCs) estabelecidos pelo *Internet Engineering Task Force* (IETF), como o RFC 2647, que define as terminologias fundamentais para a análise de desempenho de *firewalls*, e o RFC 9411, que fornece uma metodologia para o *benchmarking* de dispositivos de segurança de rede.

O RFC 9411, por exemplo, propõe uma metodologia de *benchmark* específica para *firewalls,* detalhando os testes que devem ser aplicados para avaliar as características de desempenho dos sistemas. Este documento não só descreve os tipos de testes que devem ser realizados, mas também os formatos para reportar os resultados de forma padronizada, garantindo que os dados obtidos sejam compreensíveis e possam ser comparados entre diferentes soluções.

Complementarmente, o RFC 2647 define as terminologias fundamentais para a análise de desempenho de *firewalls*, com foco nas métricas essenciais como a taxa de encaminhamento (*throughput*) e as medições orientadas a conexões, que são fundamentais para mensurar a eficiência de um *firewall*. Essas métricas ajudam a compreender a capacidade do sistema de lidar com tráfego de rede em tempo real, o que é particularmente relevante para soluções como o Raspberry Pi, que podem servir como *firewalls* de baixo custo em ambientes que exigem alto desempenho.

Em resumo, o uso de métricas padronizadas, como as propostas pelos RFCs 2647 e 9411, é relevante para garantir uma avaliação justa e precisa do desempenho de *firewalls*. Para o Raspberry Pi atuando como um *firewall*, essas métricas são essenciais para comparar seu desempenho com soluções tradicionais, verificando se ele pode atender às necessidades de segurança de redes sem sobrecarregar o orçamento. As métricas detalhadas que serão utilizadas para avaliar o desempenho, segurança, consumo de recursos e estabilidade do Raspberry Pi quando utilizado como *firewall* em redes de pequeno porte serão apresentadas no

Capítulo 3. Para alcançar resultados satisfatórios nas métricas de desempenho avaliadas, e considerando as limitações intrínsecas de hardware, técnicas de otimização se tornam essenciais, como a ordenação dinâmica de regras no *firewall*.

3 MÉTRICAS

Este capítulo dedica-se à explanação das métricas selecionadas para a análise da viabilidade técnica e do desempenho do Raspberry Pi como *firewall* de baixo custo para redes de pequeno porte. A escolha destas métricas visa abranger os aspectos cruciais que determinam a eficácia e a praticidade de um dispositivo de segurança em um cenário real. Dada a natureza do Raspberry Pi como um sistema embarcado de baixo custo, a avaliação transcende as métricas de rede tradicionais, englobando também o comportamento dos recursos de hardware sob carga.

A fundamentação para a seleção destas métricas encontra-se em padrões consolidados pela indústria e academia, como as terminologias e metodologias propostas nos RFCs 2647 e RFC 9411, que orientam a avaliação de desempenho de dispositivos de segurança de rede. Estas diretrizes foram adaptadas para contemplar as especificidades de um dispositivo com recursos computacionais inerentemente limitados, como o Raspberry Pi. O propósito é construir uma análise quantificável e rigorosa, permitindo inferências substanciadas sobre a adequação do dispositivo para a função de *firewall*.

A metodologia de avaliação proposta neste estudo, baseia-se na integração de princípios de:

- *Benchmarking* de Dispositivos de Rede: Utilizando como referência as definições e os conceitos de métricas de desempenho de rede estabelecidos em documentos como o RFC 2544, RFC 2647 e RFC 9411. Isso assegura que a compreensão das métricas de tráfego e segurança esteja alinhada com as práticas reconhecidas pela comunidade de redes
- Monitoramento de Desempenho de Sistemas Operacionais: Para as métricas relacionadas ao consumo de recursos de *hardware*, a ênfase recai sobre a importância de observar o comportamento do sistema operacional Linux, que gerencia estes recursos, e como sua utilização impacta o desempenho global do *firewall*.
- Teste de Robustez e Estabilidade de Sistemas Embarcados: As métricas de segurança, resiliência, estabilidade e escalabilidade são compreendidas à luz dos princípios de engenharia de confiabilidade e testes de estresse, que buscam aferir o comportamento do sistema sob condições adversas ou prolongadas de uso.

As métricas estão organizadas em quatro categorias principais: Desempenho de Rede, Segurança e Resiliência, Consumo de Recursos e Estabilidade e Escalabilidade.

3.1 Desempenho de Rede

As métricas de desempenho de rede permitem avaliar a capacidade do Raspberry Pi, quando empregado como *firewall*, de controlar e processar o tráfego que circula pela rede. Um desempenho inadequado nesta categoria pode introduzir gargalos, lentidão e comprometer a experiência do usuário e a funcionalidade dos serviços da rede. A Tabela 1 apresenta as principais métricas utilizadas para avaliar o comportamento do *firewall*.

Tabela 1 - Métricas para Desempenho de Rede

Métrica	Unidade	Descrição
Throughput	Mbps / Gbps	Taxa efetiva de transferência de dados que o *firewall* consegue inspecionar e encaminhar.
Latência	ms	Atraso incorrido por um pacote ao transitar pela rede, atravessando o *firewall*.
Número máximo de conexões simultâneas	conexões	Quantidade de sessões TCP ativas que o *firewall* pode gerenciar sem degradação.
Perda de Pacotes	%	Percentual de pacotes que não alcançam seu destino ao passar pelo *firewall*.
Tempo de Processamento de Pacotes	ms	Overhead temporal introduzido pelo *firewall* no encaminhamento de cada pacote individual.

Fonte: Autoria Própria

3.1.1 Throughput

O *throughput,* ou taxa de transferência efetiva, é definido como a quantidade máxima de dados por unidade de tempo que um dispositivo de rede, como um *firewall*, consegue processar e encaminhar de forma bem-sucedida entre suas interfaces. Esta métrica é expressa, comumente, em Megabits por segundo (Mbps) ou Gigabits por segundo (Gbps). A sua relevância reside no fato de que um firewall com baixo throughput pode se tornar um gargalo para a rede, limitando a velocidade de comunicação para todos os usuários e aplicações, mesmo que os *links* de internet e a infraestrutura local possuam capacidade superior. Para *firewalls*, o *throughput* é frequentemente avaliado sob diferentes condições, como tipos de tráfego (TCP vs. UDP) e tamanhos de pacote, pois estes fatores podem influenciar significativamente o desempenho do processamento de regras de filtragem e inspeção. Em dispositivos com capacidade limitada, como o Raspberry Pi, é importante saber qual throughput ele consegue manter de forma estável, pois isso determina o porte da rede que ele consegue atender sem comprometer o desempenho.

3.1.2 Latência

A latência, frequentemente mensurada como o Tempo de Ida e Volta (RTT), representa o tempo total que um pacote de dados leva para viajar de uma origem a um destino e retornar à origem, passando através do *firewall*. É uma métrica expressa em milissegundos (ms). Uma latência elevada introduzida pelo *firewall* pode degradar severamente a performance de aplicações sensíveis a atrasos, como Voz sobre IP (VoIP), videoconferências, jogos online e outras interações em tempo real. O impacto do *firewall* na latência é um indicador do *overhead* de processamento que ele adiciona ao tráfego de rede. A avaliação desta métrica considera a variação da latência sob diferentes cargas de tráfego e número de conexões ativas, pois um *firewall* sobrecarregado tende a introduzir maiores atrasos

3.1.3 Número Máximo de Conexões Simultâneas

Esta métrica refere-se à capacidade do *firewall* de rastrear e gerenciar múltiplas sessões de comunicação ativas, majoritariamente conexões TCP, ao mesmo tempo, sem que haja falhas ou uma degradação expressiva no desempenho geral. É quantificada pelo número total de conexões que o dispositivo pode manter em sua tabela de estados. Um *firewall stateful*, por exemplo, precisa alocar memória e recursos de processamento para cada conexão ativa. Atingir o limite de conexões simultâneas pode levar à recusa de novas conexões legítimas ou à instabilidade do sistema. Para um dispositivo como o Raspberry Pi, com memória e CPU limitadas, esta métrica é particularmente importante em ambientes onde há grande volume de conexões paralelas, como redes com muitos usuários navegando simultaneamente, aplicações P2P que estabelecem centenas de sessões por segundo para troca de arquivos, ou cenários com dispositivos IoT que mantêm conexões persistentes para envio contínuo de dados ao servidor.

3.1.4 Perda de Pacotes

A perda de pacotes é o percentual de pacotes de dados que, ao serem transmitidos através do *firewall,* não conseguem alcançar seu destino final. Esta métrica, expressa em porcentagem (%), é um indicador crítico da confiabilidade da rede e da capacidade do firewall de processar o tráfego sob carga. Perdas elevadas de pacotes podem ser causadas por diversos fatores, incluindo *buffers* de rede sobrecarregados no *firewall*, descarte de pacotes devido a erros de processamento, ou mesmo pela ação de regras de segurança que descartam tráfego malicioso ou não conforme. Para aplicações como *streaming* de mídia, VoIP e jogos *online*, mesmo uma pequena porcentagem de perda de pacotes pode causar degradação perceptível na qualidade do serviço. Portanto, um *firewall* eficiente deve introduzir uma perda de pacotes mínima para o tráfego legítimo.

3.1.5 Tempo de Processamento de Pacotes

O tempo de processamento de pacotes, também conhecido como *forwarding delay* ou latência por pacote, é o tempo que um pacote individual leva para ser

processado pelo *firewall*, desde o momento em que é recebido em uma interface de entrada até ser transmitido por uma interface de saída. Esta métrica é medida em milissegundos (ms) ou microssegundos (mus). Diferentemente da latência RTT, que mede o tempo de ida e volta, o tempo de processamento de pacotes foca no *overhead* computacional específico introduzido pelo dispositivo. Este *overhead* é influenciado pela complexidade das regras de *firewall*, a necessidade de inspeção de estado, tradução de endereços de rede (NAT), e outras funcionalidades de segurança. Em *firewalls* implementados em hardware com recursos limitados, como o Raspberry Pi, um tempo de processamento de pacotes elevado pode ser um indicativo de que a CPU está se tornando um gargalo, impactando diretamente o *throughput* máximo e a latência geral da rede.

3.2 Segurança e Resiliência

As métricas de segurança e resiliência avaliam a robustez do Raspberry Pi como *firewall* capaz de proteger a rede contra ameaças cibernéticas e sua capacidade de manter a integridade e a disponibilidade da rede. A Tabela 2 apresenta os principais indicadores utilizados para essa análise, destacando parâmetros que influenciam diretamente a proteção do ambiente e a capacidade do dispositivo de se manter funcional mesmo sob condições adversas.

Tabela 2 - Métricas para Segurança e Resiliência

(continua)

Métrica	Unidade / Escala	Descrição
Resistência a ataques DDoS	conexões/s	Capacidade do *firewall* de manter a disponibilidade dos serviços da rede durante um ataque DDoS.
Filtragem de tráfego malicioso	% pacotes bloqueados	Eficácia do *firewall* em identificar e bloquear tráfego indesejado ou perigoso.

Fonte: Autoria Própria

Tabela 2 - Métricas para Segurança e Resiliência

(continuação)

Métrica	Unidade / Escala	Descrição
Tempo de resposta a ataques	milissegundos (ms)	Intervalo entre a detecção de uma ameaça e a aplicação de contramedidas pelo *firewall.*
Eficiência na aplicação de regras	% de variação no *throughput*	Impacto no desempenho da rede causado pela carga de processamento das regras do *firewall*.

Fonte: Autoria Própria

3.2.1 Resistência a Ataques de Negação de serviço

A resistência a ataques de Negação de Serviço Distribuída (DDoS) mede a capacidade do *firewall* de suportar e mitigar os efeitos de ataques que visam exaurir seus recursos (CPU, memória, tabela de conexões) ou a largura de banda da rede, tornando os serviços indisponíveis para usuários legítimos. Esta métrica pode ser quantificada pela máxima taxa de tráfego de ataque (em conexões por segundo, pacotes por segundo ou Mbps) que o *firewall* consegue suportar antes de apresentar uma degradação significativa dos serviços protegidos ou falha do próprio dispositivo. Para um *firewall* em um dispositivo de recursos limitados como o Raspberry Pi, a resistência a DDoS é um desafio considerável, pois mesmo volumes moderados de tráfego de ataque podem sobrecarregar rapidamente o sistema. Avaliar esta métrica envolve observar a estabilidade do *firewall* e a continuidade do fluxo de tráfego legítimo enquanto o ataque está em curso.

3.2.2 Capacidade de Filtragem de Tráfego Malicioso

Esta métrica avalia a precisão e a eficácia do *firewall* em identificar e bloquear tráfego que é considerado malicioso ou que viola as políticas de segurança estabelecidas. É tipicamente expressa como a porcentagem de pacotes ou fluxos de

tráfego malicioso conhecidos que são corretamente bloqueados pelo *firewall*. O tráfego malicioso pode incluir tentativas de exploração de vulnerabilidades, varreduras de portas, comunicação com domínios maliciosos conhecidos, ou pacotes que correspondem a assinaturas de *malware*. A capacidade de filtragem depende da qualidade e da abrangência das regras configuradas, da inteligência de ameaças utilizada e dos mecanismos de inspeção do *firewall.* Para o Raspberry Pi, a complexidade da filtragem que pode ser implementada impactará diretamente esta métrica, devido às limitações de processamento.

3.2.3 Tempo de Resposta a Ataques

O tempo de resposta a ataques refere-se ao intervalo entre o momento em que uma atividade maliciosa é detectada e o momento em que o *firewall* efetivamente implementa uma ação de mitigação para bloquear ou neutralizar a ameaça. Medido em segundos ou milissegundos, um tempo de resposta baixo é crucial para minimizar a janela de oportunidade para um atacante causar danos. Este tempo pode ser influenciado pela capacidade de processamento do *firewall* para analisar *logs* ou alertas, pela eficiência com que novas regras de bloqueio podem ser aplicadas dinamicamente, e pela integração com outros sistemas de segurança. Em *firewalls* mais simples, a "resposta" pode ser a aplicação de regras predefinidas que bloqueiam o tráfego anômalo assim que ele é identificado.

3.2.4 Eficiência na Aplicação de Regras

A eficiência na aplicação de regras mede o impacto que o conjunto de regras do *firewall* (sua quantidade, complexidade e ordem) tem sobre o desempenho da rede, particularmente no *throughput e* na latência. Um grande número de regras complexas, ou regras mal otimizadas, pode consumir excessivos recursos de CPU e memória para processar cada pacote, resultando em uma diminuição da taxa de transferência e aumento da latência. Esta métrica é avaliada comparando o desempenho da rede com diferentes conjuntos de regras ativas versus um cenário base com o mínimo de regras possível, quantificando a degradação percentual.

3.3 Consumo de Recursos

As métricas de consumo de recursos ajudam a medir a eficiência do Raspberry Pi quando utilizado como *firewall,* levando em conta as limitações do seu hardware. Elas indicam o quão bem o dispositivo utiliza seus recursos computacionais (CPU, memória) e gerencia aspectos físicos (temperatura) sob diferentes cargas de trabalho, o que tem impacto direto na estabilidade e no desempenho sustentado. A Tabela 3 apresenta as principais métricas consideradas nessa avaliação, contemplando o uso de CPU, a alocação de memória RAM e a temperatura operacional do processador.

Tabela 3 - Métricas para Consumo de Recursos

Métrica	Unidade	Descrição
Uso de CPU	%	Percentual do tempo de processamento da CPU utilizado pelas operações do *firewall* e do sistema.
Uso de Memória RAM (*Random Access Memory*)	MB / %	Quantidade de memória RAM consumida pelos processos do *firewall* e pelo sistema operacional.
Temperatura do Processador	°C	Temperatura operacional da CPU, indicativa de estresse térmico e potencial throttling.

Fonte: Autoria Própria

3.3.1 Uso de CPU

O uso de CPU quantifica a carga sobre a Unidade Central de Processamento do Raspberry Pi, expressa como uma porcentagem do tempo total de processamento disponível. Um *firewall*, ao inspecionar pacotes, manter tabelas de estado, aplicar regras e realizar outras funções de segurança, consome ciclos de CPU. Um uso de CPU consistentemente alto (acima de 80-90%) pode indicar que o dispositivo está operando próximo ao seu limite de capacidade, o que pode levar a um aumento da latência, redução do *throughput* e, em casos extremos, instabilidade do sistema. O monitoramento do uso de CPU sob diferentes cargas de tráfego e

com diferentes conjuntos de regras de *firewall* é essencial para identificar gargalos de desempenho e entender os limites operacionais do Raspberry Pi como *firewall*.

3.3.2 Uso de Memória RAM

O uso de Memória de Acesso Aleatório (RAM) refere-se à quantidade de memória física (geralmente medida em Megabytes ou como percentual do total disponível) que está sendo utilizada pelo sistema operacional e pelos processos do firewall em um dado momento. Funções do firewall, como o armazenamento da tabela de conexões (*state table*), *buffers* para pacotes de rede, e os próprios processos de *software* do *firewall* consomem RAM. Um consumo excessivo de memória pode levar à paginação (*swapping*), se configurada, o que degrada severamente o desempenho, ou ao esgotamento da memória, resultando em falhas de processos (incluindo o próprio *firewall)* ou do sistema (*Out-Of-Memory killer*). Para um dispositivo como o Raspberry Pi, que possui uma quantidade limitada de RAM, o monitoramento desta métrica é importante para garantir a estabilidade.

3.3.3 Temperatura do Processador

A temperatura do processador é uma métrica física, medida em graus Celsius (°C), que indica o quão aquecido o principal componente de processamento do Raspberry Pi se encontra durante a operação. Cargas de trabalho intensas, como o processamento contínuo de alto volume de tráfego de rede ou a execução de regras de *firewall* complexas, aumentam a atividade da CPU, o que, por sua vez, gera calor. Se a temperatura exceder limiares críticos (tipicamente em torno de 80-85°C para o Raspberry Pi), mecanismos de proteção como o *thermal throttling* são ativados, reduzindo a velocidade do *clock* da CPU para diminuir a geração de calor. Isso, no entanto, resulta em uma queda no desempenho. O monitoramento da temperatura é importante para garantir que o Raspberry Pi opere dentro de limites seguros, evitando danos ao hardware a longo prazo e assegurando um desempenho consistente.

3.4 Estabilidade e Escalabilidade

As métricas de estabilidade e escalabilidade são cruciais para avaliar a confiabilidade do Raspberry Pi como *firewall* em operação contínua e sua capacidade de se adaptar a um aumento gradual na demanda da rede ou no número de dispositivos conectados. A Tabela 4 apresenta essas métricas e suas respectivas unidades de análise.

Tabela 4 - Métricas para Estabilidade e Escalabilidade

Métrica	Unidade	Descrição
Tempo de uptime sem falhas	Horas (h) / Dias (d)	Duração contínua da operação do *firewall* sem reinicializações inesperadas ou falhas críticas do sistema.
Degradação de desempenho ao longo do tempo	Mbps / % variação	Variação (geralmente negativa) nas métricas de desempenho (*throughput*, latência) após longos períodos de operação.
Capacidade em redes maiores	Nº de dispositivos	Habilidade do *firewall* de gerenciar um número crescente de dispositivos e tráfego associado sem severa degradação.

Fonte: Autoria Própria

3.4.1 Tempo de Uptime sem Falhas

O tempo de *uptime* (tempo em atividade) sem falhas é uma medida direta da estabilidade e confiabilidade de um sistema. Representa o período contínuo durante o qual o Raspberry Pi, atuando como *firewall*, opera conforme o esperado, sem interrupções não planejadas, como travamentos do sistema, reinicializações espontâneas ou falhas nos serviços essenciais do *firewall.* Um tempo de *uptime* elevado, medido em horas, dias ou mesmo meses, indica um sistema robusto e

confiável, o que é um requisito para um dispositivo de segurança que precisa proteger a rede continuamente. Falhas podem ser causadas por bugs de *software*, esgotamento de recursos, superaquecimento ou problemas de *hardware*.

3.4.2 Degradação de Desempenho ao Longo do Tempo

Esta métrica avalia se e como o desempenho do *firewall* (em termos de *throughput*, latência, tempo de processamento de pacotes etc.) se altera após o sistema operar continuamente por longos períodos (24 horas, uma semana). A degradação pode ser quantificada como uma variação percentual ou absoluta nas métricas de desempenho em comparação com os valores medidos no início do período de teste. Causas comuns para degradação de desempenho ao longo do tempo incluem vazamentos de memória (*memory leaks*), fragmentação de recursos, acúmulo de estados obsoletos na tabela de conexões, ou até mesmo o impacto gradual de *thermal throttling* se o sistema de refrigeração for inadequado para operação contínua sob carga. Identificar e quantificar essa degradação é chave para entender a sustentabilidade da solução de firewall a longo prazo.

3.4.3 Capacidade para Redes Maiores

A capacidade para redes maiores, ou escalabilidade, refere-se à habilidade do *firewall* baseado no Raspberry Pi de lidar com um aumento no número de dispositivos conectados à rede e, consequentemente, com o volume de tráfego e o número de conexões simultâneas, sem que haja uma queda abrupta e severa no desempenho. Esta métrica é para determinar se a solução é viável não apenas para uma rede pequena e estática, mas também para ambientes que podem crescer. A avaliação da escalabilidade envolve testar o *firewall* sob cargas crescentes que simulam a adição de mais dispositivos, observando como métricas chave (*throughput*, latência, uso de CPU/memória) são afetadas e até que ponto o sistema consegue manter um nível de serviço aceitável. As limitações de *hardware* do Raspberry Pi naturalmente impõem um teto para sua escalabilidade.

4 IMPLEMENTAÇÃO E CONFIGURAÇÃO DO AMBIENTE DE *FIREWALL*

O presente capítulo detalha o processo de concepção e implementação do ambiente de *firewall.* O objetivo é demonstrar que um conjunto de ferramentas *open-source*, executado em *hardware* de baixo custo como o Raspberry Pi 5, pode atender aos requisitos de segurança de uma pequena ou média empresa, aplicando-se ao cenário da empresa fictícia "Conecta Soluções". A abordagem traduz os desafios teóricos discutidos anteriormente, como a necessidade de soluções acessíveis e mecanismos de defesa robustos, em uma solução técnica funcional, cuja eficácia será validada por meio de testes. A implementação foi guiada para que o sistema final ofereça controles de segurança comparáveis aos de *firewalls* corporativos comerciais, porém com um custo de aquisição significativamente menor e com a transparência de soluções de código aberto.

4.1 Visão Geral da Implementação

Esta seção apresenta a concepção do ambiente de *firewall* proposto, detalhando o processo de implementação prática da solução baseada no Raspberry Pi. O objetivo é demonstrar, de forma empírica, como a integração de ferramentas *open-source* em *hardware* de baixo custo pode atender às necessidades de segurança de uma pequena ou média empresa, conforme o perfil da empresa fictícia "Conecta Soluções". A implementação busca replicar as funcionalidades centrais de um *Next-Generation Firewall* (NGFW). Diferente de *firewalls* tradicionais que operam apenas com base em portas e endereços IP, um NGFW incorpora capacidades avançadas como a *Deep Packet Inspection* (DPI), controle de acesso em nível de aplicação e sistemas de prevenção de intrusão. O presente trabalho visa construir uma solução que mimetize essas funcionalidades, mantendo, contudo, a aderência a restrições orçamentárias e a total transparência proporcionada pelo código aberto.

4.1.1 Cenário de Aplicação

Para validar a solução em um contexto realista, foi definido um cenário de aplicação baseado na "Conecta Soluções", uma empresa brasileira fictícia de pequeno a médio porte com cerca de 80 funcionários. Atuando no setor de

consultoria em tecnologia, a empresa adota um modelo de trabalho híbrido, o que exige acesso remoto seguro para parte de seus colaboradores. A empresa gerencia dados sensíveis, como informações de clientes e propriedade intelectual, demandando conformidade com a Lei Geral de Proteção de Dados (LGPD). O principal desafio de negócio é fortalecer sua segurança digital com um orçamento limitado, o que justifica a análise de uma alternativa de baixo custo aos firewalls comerciais tradicionais, um desafio comum para PMEs.

4.1.2 Política de Segurança e Requisitos Funcionais

Toda a arquitetura do firewall foi projetada para atender a uma política de segurança da informação específica, criada para proteger os ativos da "Conecta Soluções". As diretrizes dessa política e sua tradução em requisitos funcionais para o *firewall* são descritas a seguir. A política de segurança foca em cinco pilares principais para garantir a proteção dos dados e a continuidade das operações, conforme detalhado na Tabela 5.

Tabela 5 - Diretrizes da Política de Segurança da Informação

(continua)

Diretriz da Política	Descrição / Requisitos
Controle de Acesso	O acesso à rede e aos recursos internos deve seguir o princípio do menor privilégio. Todo acesso remoto deve ser realizado através de um canal seguro e criptografado. Deve existir uma rede para visitantes totalmente isolada da rede corporativa.
Proteção de Perímetro	O firewall deve filtrar e monitorar todo o tráfego de entrada e saída. O acesso à internet deve ser controlado, bloqueando sites que representem risco ou sejam inadequados ao ambiente de trabalho.

Fonte: Autoria Própria

Tabela 5 - Diretrizes da Política de Segurança da Informação

(continuação)

Diretriz da Política	Descrição / Requisitos
Prevenção contra Ameaças	A rede deve ser monitorada para detectar e bloquear tentativas de invasão, como varreduras de portas e tráfego malicioso conhecido. Downloads de arquivos executáveis de fontes não confiáveis devem ser bloqueados.
Uso Aceitável	O acesso a categorias de sites que afetam a produtividade, como redes sociais e streaming, pode ser restringido durante o horário de expediente.
Registro e Auditoria	Todos os eventos de rede relevantes, como tentativas de acesso bloqueadas, sites acessados e atividade da VPN, devem ser registrados (logs) e armazenados para fins de auditoria e análise de incidentes.

Fonte: Autoria Própria

4.1.3 Objetivos da montagem do ambiente

A montagem do ambiente experimental foi orientada por objetivos específicos, que desdobram o propósito geral de avaliar a viabilidade do Raspberry Pi 5 como um *firewall* de baixo custo em um cenário prático e controlado. Os objetivos primários foram:

- Validar a viabilidade técnica da solução: Avaliar se o hardware do Raspberry Pi 5, combinado com as ferramentas open-source selecionadas, é capaz de prover as funcionalidades de um *firewall* corporativo moderno. Isso inclui verificar sua capacidade de realizar inspeção de tráfego, segmentação de rede, filtragem e monitoramento, conforme os requisitos estipulados pela política de segurança da empresa fictícia "Conecta Soluções".

- Replicar um cenário de uso empresarial realista: Construir um ambiente de teste que simule as condições de uma rede de pequena ou média empresa para verificar não apenas as funcionalidades de forma isolada, mas também a sua integração e desempenho sob condições de uso realistas. Os requisitos funcionais a serem replicados e validados incluem:
  - *Deep Packet Inspection* (DPI) para análise de tráfego em nível de aplicação;
  - Segmentação lógica da rede em múltiplas zonas de segurança, como *Local Area Network* (LAN), *Demilitarized Zone* (DMZ), Rede de Visitantes e Servidores Privados;
  - Filtragem web com base em categorias e listas de bloqueio.
  - Suporte a acesso remoto seguro por VPN para colaboradores em regime de trabalho híbrido.
  - Registro (*logging*) centralizado de eventos de segurança e monitoramento de tráfego em tempo real para fins de auditoria e análise de incidentes.

4.1.4 Critérios de seleção das ferramentas

A seleção das ferramentas de software foi um processo criterioso, guiado pela necessidade de equilibrar estabilidade funcional, desempenho em hardware com recursos limitados e simplicidade de administração, fatores críticos para o cenário de uma pequena empresa sem uma equipe de segurança dedicada. A escolha de cada componente considerou um conjunto de requisitos técnicos e operacionais, garantindo que a solução final fosse coesa, segura e alinhada aos objetivos do projeto. Os principais critérios para a seleção foram:

- Adequação funcional: capacidade de atender às funcionalidades essenciais, intermediárias e avançadas (DPI, IDS/IPS, proxy, VPN).
- Desempenho: Evidências de operação estável e desempenho otimizado na arquitetura *Advanced RISC Machine* (ARM) de 64 bits, utilizada pelo Raspberry Pi 5.
- Maturidade e suporte comunitário: ampla base de usuários, atualizações frequentes e documentação.

- Integração simplificada: compatibilidade entre ferramentas por meio de *hooks* de firewall, APIs e integração de *logs*.
- Licenciamento open-source: uso de licenças permissivas (GPL, BSD, Apache) para assegurar independência de fornecedor e ausência de custos.
- Baixa superfície de ataque: histórico de segurança consistente e *roadmap* ativo de correções.

Com base nesses critérios, foram selecionadas as ferramentas. A Tabela 6 organiza essas ferramentas, descrevendo a função desempenhada por cada uma delas no ambiente de firewall e apresentando as justificativas que motivaram sua escolha.

Tabela 6 - Ferramentas de Software Selecionadas e Justificativas

(continua)

Ferramenta	Função Principal	Justificativa da Escolha
nftables	Firewall de Estado e NAT	Sucessor moderno do iptables, com melhor desempenho e sintaxe simplificada, ideal para filtragem stateful.
Suricata	IDS/IPS e DPI	Sistema de alto desempenho com capacidade de DPI e integração com feeds de ameaças para bloqueio ativo de ataques.
Squid + SquidGuard	Proxy Web e Filtro de Conteúdo	Combinação para inspeção de tráfego web, bloqueio por categorias e aplicação de políticas de uso.
WireGuard	VPN para Acesso Remoto	Solução VPN moderna, leve e segura, com desempenho superior, ideal para hardware com recursos limitados.

Fonte: Autoria Própria

Tabela 6 - Ferramentas de Software Selecionadas e Justificativas

(continuação)

Ferramenta	Função Principal	Justificativa da Escolha
Loki + Promtail + Grafana	Coleta e Visualização de Logs	Pilha de logging moderna e leve, adequada para os recursos do Raspberry Pi, permitindo a criação de dashboards centralizados.
ntopng	Monitoramento de Tráfego	Oferece visualização em tempo real do uso da rede com uma interface web intuitiva, essencial para a análise de padrões.

Fonte: Autoria Própria

4.2 Infraestrutura de Hardware e Rede

A validação de uma solução de segurança de rede exige a construção de um ambiente de testes que simule, com a maior fidelidade possível, as condições de uma infraestrutura corporativa real. Esta seção detalha os componentes de *hardware*, a topologia de rede e a arquitetura lógica implementadas para avaliar o desempenho do Raspberry Pi como um *firewall* para a empresa fictícia "Conecta Soluções". A infraestrutura foi projetada para ser robusta, segmentada e controlada, permitindo a execução de testes sistemáticos e a coleta de métricas confiáveis.

4.2.1 Especificações do Raspberry Pi utilizado

O hardware central da solução, que atua como o *appliance* de *firewall*, é um Raspberry Pi 5. A escolha deste modelo foi motivada por ser a geração mais recente e performática da plataforma no momento da elaboração deste trabalho, oferecendo um avanço significativo em capacidade de processamento e E/S (Entrada/Saída) em comparação com seus predecessores. As especificações técnicas do dispositivo utilizado são:

- Processador (CPU): Broadcom BCM2712, 2.4GHz quad-core 64-bit Arm Cortex-A76. A alta capacidade de processamento é fundamental para

tarefas intensivas como *Deep Packet Inspection* (DPI) e a operação do Intrusion Prevention System (IPS).

- Memória RAM: 4GB. A memória é importante para gerenciar a tabela de estado de conexões (*stateful inspection*), execução das regras do Suricata e funcionamento do *proxy*, reduzindo a dependência de *swap*, que poderia impactar negativamente o desempenho.
- Armazenamento: Cartão microSD de 64GB, Classe A2, para o sistema operacional e armazenamento de *logs*.
- Interfaces de Rede: determinam como o tráfego será recebido, inspecionado e encaminhado. O Raspberry Pi utilizado conta com:
  - Porta Gigabit Ethernet integrada (eth0): designada como interface *Wide Area Network* (WAN), conectada à internet.
  - Adaptador USB 3.0 para Gigabit Ethernet (eth1): Adicionado para prover uma segunda interface física, utilizada como *Local Area Network* (LAN), conectada à rede interna simulada.
  - Wi-Fi 5 (802.11ac) Integrado (wlan0): Utilizado para a criação da rede de visitantes (Guest), garantindo seu isolamento físico e lógico.

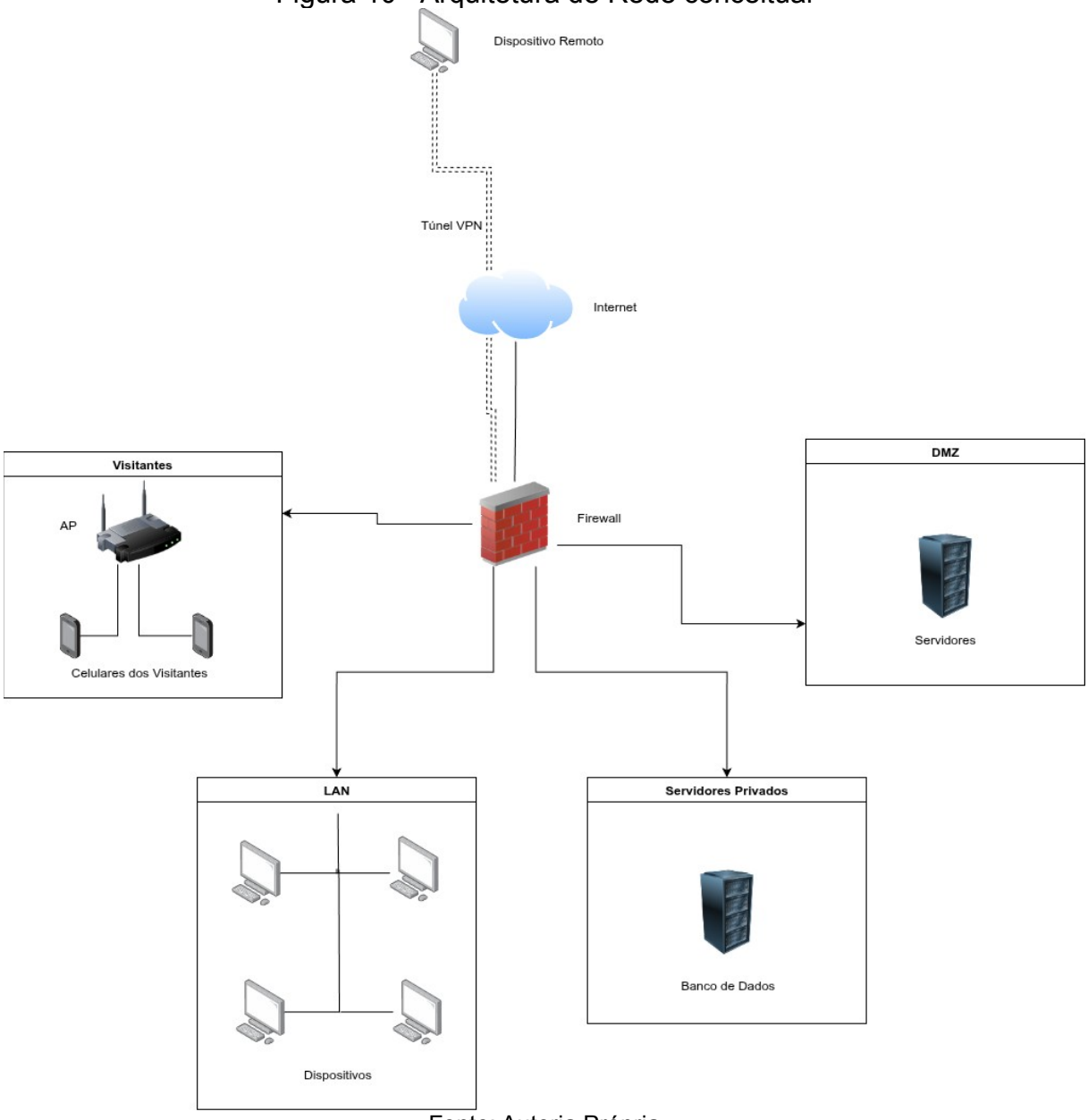
4.2.2 Topologia da Rede Implementada

A topologia da rede foi desenhada para posicionar o Raspberry Pi como um *gateway* de perímetro, centralizando todo o fluxo de dados entre a rede interna da empresa e a internet. Essa arquitetura reflete o cenário padrão de implantação de um *firewall* corporativo, onde o dispositivo atua como o único ponto de controle e inspeção para todo o tráfego que cruza as fronteiras da rede.

Na topologia implementada, o firewall é o elemento central que segmenta a infraestrutura em múltiplas zonas de segurança, como a Rede Local (LAN), a Zona Desmilitarizada (DMZ), a rede de Servidores Privados e a rede de Visitantes. O tráfego originado da internet chega pela interface WAN, é processado pelas regras de *firewall*, IPS e proxy, e então encaminhado para a interface de destino apropriada. A topologia também contempla o acesso seguro para dispositivos remotos por meio de um túnel de VPN. De forma análoga, todo o tráfego de saída das redes internas passa obrigatoriamente pelo Raspberry Pi antes de alcançar a internet, garantindo

que as políticas de segurança sejam aplicadas em ambas as direções. A topologia implementada está ilustrada conceitualmente na Figura 10.

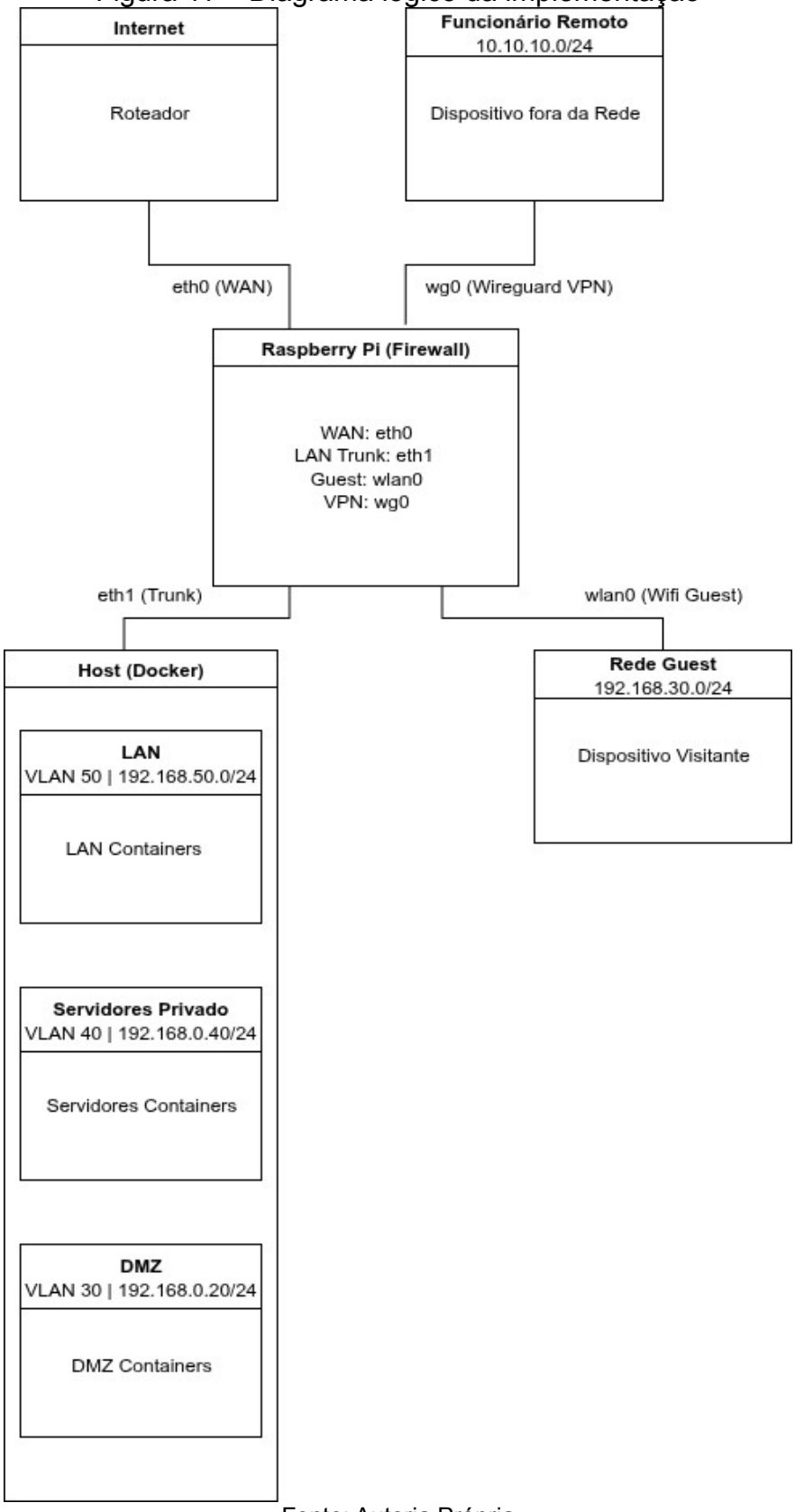
Figura 10 - Arquitetura de Rede conceitual



Fonte: Autoria Própria

Para transformar esse arranjo conceitual em uma implementação real, foi desenvolvida uma topologia lógica no Raspberry Pi, detalhada na Figura 11. Diferentemente da arquitetura geral apresentada anteriormente, esta figura descreve como cada interface física e virtual foi configurada para dar suporte às zonas de segurança.

Figura 11 - Diagrama lógico da implementação



Fonte: Autoria Própria

No modelo implementado, o Raspberry Pi opera com as seguintes interfaces:

- Interface WAN (eth0): Conectada à internet, servindo como o ponto de entrada e saída para todo o tráfego externo.
- Interface LAN Trunk (eth1): Configurada como uma porta trunk, é responsável por transportar o tráfego de múltiplas redes virtuais (VLANs) que segmentam a rede interna, incluindo a LAN (VLAN 50), Servidores Privados (VLAN 40) e a DMZ (VLAN 30).
- Interface Guest (wlan0): A interface Wi-Fi integrada é utilizada para criar uma rede de visitantes totalmente isolada das redes corporativas, oferecendo apenas acesso à internet.
- Interface VPN (wg0): Uma interface virtual criada pelo WireGuard para prover acesso remoto seguro aos funcionários.

Este modelo caracteriza um "*firewall* na borda", garante que todo o tráfego, seja ele de entrada, saída ou entre as zonas de segurança internas, passe obrigatoriamente pelo Raspberry Pi. Tal centralização permite a aplicação consistente das políticas de segurança, a inspeção de tráfego pelo Sistema de Prevenção de Intrusão (IPS) e a filtragem de conteúdo web, proporcionando um controle granular e uma visibilidade completa sobre as comunicações da rede.

4.2.3 Segmentação de rede e VLANs

A segmentação da rede é uma das bases da estratégia de segurança adotada, transformando a política da "Conecta Soluções" em divisões lógicas que ajudam a proteger os ativos de informação. Para atender à exigência de isolamento de recursos críticos e aplicar o princípio do menor privilégio, a rede interna foi dividida em múltiplas zonas de segurança com diferentes níveis de confiança.

Para alcançar o isolamento necessário entre os diferentes setores da empresa, como a rede corporativa, servidores e acesso para visitantes,sem a necessidade de múltiplos equipamentos físicos, foi implementada a segmentação lógica por meio de *Virtual Local Area Networks* (VLANs). Todo o tráfego segmentado é transportado pela interface física eth1, configurada como uma porta trunk. No Raspberry Pi, foram criadas subinterfaces virtuais atreladas a cada VLAN, permitindo que o firewall aplique políticas de segurança distintas para cada segmento de rede de forma centralizada. As zonas de segurança definidas,

juntamente com suas respectivas configurações de rede e interfaces lógicas no firewall, estão detalhadas na Tabela 7.

Tabela 7 - Zonas de Segurança e Segmentação da Rede

Zona de Segurança	Interface Lógica (no Pi)	Sub-rede IP	Finalidade e Política de Segurança
LAN Corporativa	eth1	192.168.50.0/24	Rede principal para estações de trabalho. Possui acesso à internet (filtrado) e aos servidores da DMZ.
DMZ	eth1.20 (VLAN 20)	192.168.20.0/24	Zona Desmilitarizada para servidores com exposição à internet (ex: Servidor Web). Isolada da LAN, só permite tráfego de entrada em portas específicas.
Servidores Privados	eth1.40 (VLAN 40)	192.168.40.0/24	Rede para servidores críticos e bancos de dados, sem acesso direto da internet. O acesso é restrito e controlado a partir da LAN.
Rede de Visitantes	wlan0	192.168.30.0/24	Rede sem fio para visitantes. Oferece apenas acesso à internet e é totalmente isolada das demais redes corporativas.
Rede VPN	wg0	10.10.10.0/24	Interface virtual para clientes de VPN (WireGuard). Concede acesso remoto seguro à LAN e à DMZ.

Fonte: Autoria Própria

Para demonstrar a configuração prática dessas interfaces no sistema operacional do *firewall*, a Figura 12 exibe a saída dos comandos *nmcli* (uma ferramenta de linha de comando para controlar o *NetworkManager* e relatar o status da rede, usada para gerenciar conexões de rede e status do dispositivo*)* confirma a materialização da arquitetura lógica, mostrando as subinterfaces virtuais (eth1.20, eth1.40 e eth1.50) atreladas à porta trunk eth1, bem como o status operacional das demais interfaces de rede, como a wlan0 para a rede de visitantes e a wg0 para a VPN.

Figura 12 - Status das Interfaces e Conexões de VLAN no Firewall Raspberry Pi

```
jeff@firewall-pi:~ $ nmcli connection show
NAME                UUID                                  TYPE       DEVICE
Wired connection 1  e0e99496-88e2-36bd-bdd7-4beefe45c5ab  ethernet   eth0
ethPi               34dd7934-d2b3-44ef-b1a4-09b11b305ca1  ethernet   eth1
guest-wifi          3ba77f44-bacc-448c-abb0-9e11c7ea30a6  wifi       wlan0
lo                  86e3b540-521d-42e9-8ac2-b69cf06f292b  loopback   lo
vlan20              a1284037-e90f-40b7-acb2-08684fb5ae6a  vlan       eth1.20
vlan40              8c8de9c0-2674-49b3-ae5a-aa7770dc8496  vlan       eth1.40
wg0                 ac409fd1-b054-4489-bc7e-afaee48d75e4  wireguard  wg0
jeff@firewall-pi:~ $ nmcli device status
DEVICE         TYPE       STATE                   CONNECTION
eth0           ethernet   connected               Wired connection 1
eth1           ethernet   connected               ethPi
lo             loopback   connected (externally)  lo
eth1.20        vlan       connected               vlan20
eth1.40        vlan       connected               vlan40
wlan0          wifi       connected               guest-wifi
wg0            wireguard  connected (externally)  wg0
```

Fonte: Autoria Própria

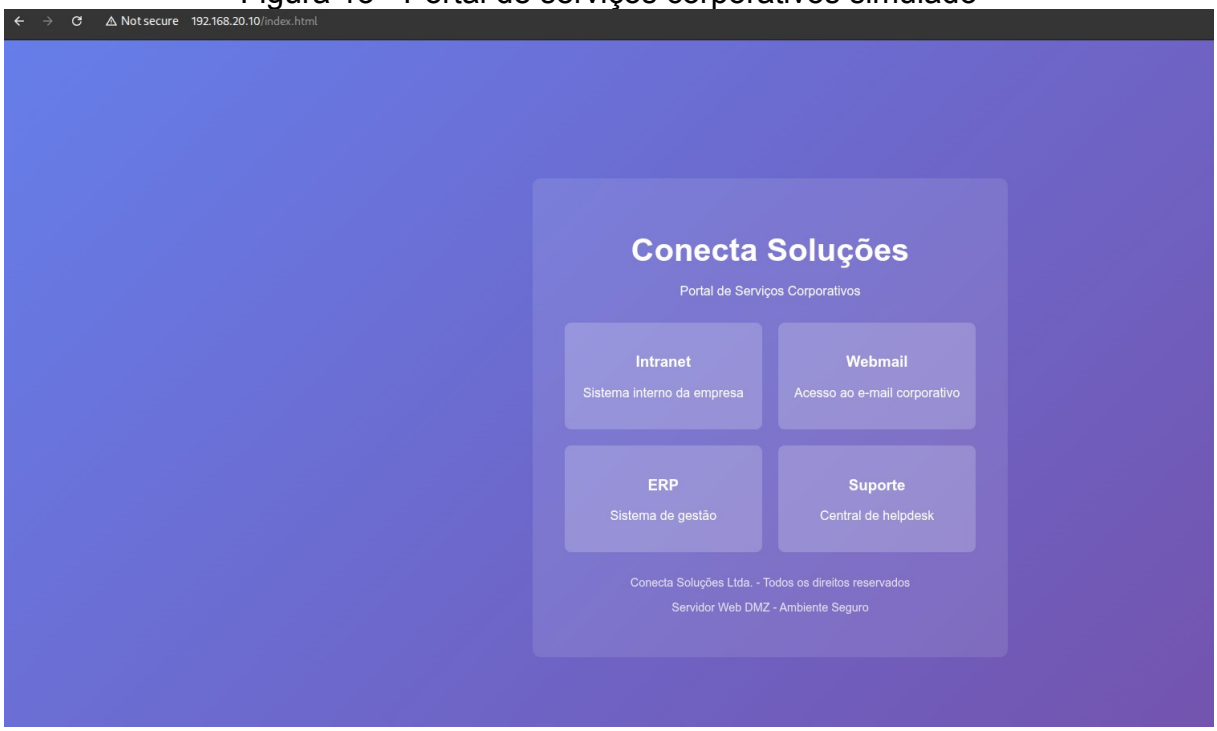
Essa arquitetura de segmentação robusta assegura que um eventual comprometimento em uma zona de menor confiança, como a DMZ ou a rede de visitantes, não se propague lateralmente para as redes internas críticas (LAN e Servidores Privados), reduzindo a superfície de ataque e fortalecendo a postura de segurança da organização.

4.2.4 Equipamentos auxiliares e conectividade

Além do Raspberry Pi, o ambiente de laboratório contou com os seguintes equipamentos para simular a infraestrutura completa e realizar os testes.

Host de Simulação: Utilizou-se um computador desktop com sistema operacional Linux como anfitrião (*host*) para a "Rede Interna da Empresa". Nele, a plataforma Docker foi empregada para instanciar contêineres que simulam os diversos ativos corporativos, como estações de trabalho na LAN e servidores (Web na DMZ, banco de dados na rede de Servidores Privados). Este *host* foi conectado diretamente à interface eth1 do Raspberry Pi, que foi configurada para operar como uma porta trunk. Por meio da criação de VLANs e interfaces MacVLAN, foi possível segmentar o tráfego em redes lógicas distintas sobre uma única conexão física, espelhando a arquitetura definida. Para validar a funcionalidade da zona desmilitarizada (DMZ), foi instanciado um contêiner web simulando o portal de serviços da 'Conecta Soluções'. Este serviço serve como alvo para os testes de acesso externo e validação das regras de segmentação, conforme ilustrado na Figura 13.

Figura 13 - Portal de serviços corporativos simulado



Fonte: Autoria Própria

A Figura 14 detalha a materialização desta arquitetura de conectividade no *host* de simulação. A saída dos comandos nmcli demonstra que a interface física “enp3s0f3u4” foi configurada como base para a criação de subinterfaces virtuais de VLAN (enp3s0f3u4.20 e enp3s0f3u4.40), permitindo que o sistema operacional do

host decodifique o tráfego dos segmentos de rede da DMZ (VLAN 20) e dos Servidores Privados (VLAN 40). Em seguida, a saída do comando docker network ls revela como o Docker utiliza o *driver* macvlan para criar redes como dmz_net, lan_macvlan e server_net. Esse *driver* permite que os contêineres sejam diretamente conectados a uma subinterface de VLAN do *host*, recebendo um endereço IP próprio dentro daquele segmento e se comportando como dispositivos fisicamente presentes na rede correspondente. Em suma, o *host* segrega o tráfego das VLANs em interfaces lógicas, e o Docker, por meio do macvlan, conecta os contêineres a essas interfaces, inserindo-os efetivamente nos segmentos de rede corretos para serem gerenciados pelo firewall.

Figura 14 - Interfaces VLAN no Host e Redes Macvlan no Docker

```
jeff@fedora:~$ nmcli device status
DEVICE          TYPE      STATE                   CONNECTION
enp3s0f3u4      ethernet  connected               ethPi
enp3s0f3u4.20   vlan      connected               vlan20
enp3s0f3u4.40   vlan      connected               vlan40
lo              loopback  connected (externally) lo
docker0         bridge    connected (externally) docker0
wlp2s0          wifi      unavailable             --
jeff@fedora:~$ nmcli connection show
NAME       UUID                                  TYPE      DEVICE
ethPi      d8364340-7c7c-4d4b-b74a-01d9cd5dd0b5  ethernet  enp3s0f3u4
vlan20     d8aefd6f-dce4-4e5d-8584-4a1e0f29d220  vlan      enp3s0f3u4.20
vlan40     c5adec70-d55e-42a6-ae1b-4305e920e9d1  vlan      enp3s0f3u4.40
lo         69876b8f-decc-4138-ad71-24e0f282b77e  loopback  lo
docker0    8e67fe5c-63fd-4aa6-b04f-f37c810f31aa  bridge    docker0
ethRouter  84fbad38-3b84-4e03-9b4b-840c05eeda16  ethernet  --
NET        1c8a9f2c-314c-4967-ab43-99b66091b3b2  wifi      --
jeff@fedora:~$ docker network ls
NETWORK ID     NAME          DRIVER    SCOPE
5cb0154cb45a   bridge        bridge    local
05d14dc11383   dmz_net       macvlan   local
216efc2769e3   host          host      local
7de55f7b410c   lan_macvlan   macvlan   local
f7826a89ba7c   none          null      local
8b7fa1090762   server_net    macvlan   local
```

Fonte: Autoria Própria

A Figura 15 detalha a inspeção da rede dmz_net, confirmando sua associação direta à subinterface física correspondente e o endereçamento IP isolado, permitindo que os contêineres atuem como hosts independentes na rede.

Figura 15 - Configuração de rede dos contêineres Macvlan

```
jeff@fedora:~$ docker network ls
NETWORK ID     NAME          DRIVER    SCOPE
3cd58d099535   bridge        bridge    local
fabbefc5dfb1   dmz_net       macvlan   local
216efc2769e3   host          host      local
7de55f7b410c   lan_macvlan   macvlan   local
f7826a89ba7c   none          null      local
8b7fa1090762   server_net    macvlan   local
jeff@fedora:~$ docker network inspect dmz_net
[
    {
        "Name": "dmz_net",
        "Id": "fabbefc5dfb147039f1e56415a3bf8b9bb63d336618b953b20aadfa97bbdcf7f",
        "Created": "2025-09-30T23:48:59.155669788-03:00",
        "Scope": "local",
        "Driver": "macvlan",
        "EnableIPv4": true,
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": {},
            "Config": [
                {
                    "Subnet": "192.168.20.0/24",
                    "Gateway": "192.168.20.1"
                }
            ]
        },
        "Internal": false,
        "Attachable": false,
        "Ingress": false,
        "ConfigFrom": {
            "Network": ""
        },
        "ConfigOnly": false,
        "Containers": {},
        "Options": {
            "parent": "enp3s0f3u4.20"
        },
        "Labels": {}
    }
]
jeff@fedora:~$
```

Fonte: Autoria Própria

Roteador de Acesso à Internet: Um roteador doméstico padrão foi utilizado para simular a "Internet". Este dispositivo proveu a conectividade externa e o serviço de DHCP para a interface WAN (eth0) do Raspberry Pi, permitindo que o *firewall* obtivesse um endereço IP público (no contexto do laboratório) e acesso ao exterior.

Cliente de Teste Remoto: Um segundo dispositivo foi conectado à rede do roteador de acesso, posicionando-o externamente ao *firewall*. Sua finalidade foi simular o acesso de um funcionário remoto para validar a funcionalidade da VPN e também atuar como um ponto de origem para testes de segurança, como tentativas de varredura de portas contra a interface WAN.

A conectividade física foi estabelecida da seguinte forma: o roteador foi conectado à porta eth0 do Raspberry Pi, e o host de simulação foi conectado à porta eth1. Essa disposição simples, porém eficaz, garantiu que todo o tráfego gerado pelos contêineres Docker (simulando a empresa) fosse forçado a passar pelo Raspberry Pi para qualquer comunicação externa, permitindo a validação de todas as funcionalidades do *firewall*.

4.3 Sistema Operacional e Preparação do Ambiente Base

Para que um firewall funcione bem, ele precisa começar com um sistema operacional seguro, ajustado corretamente e preparado para suportar suas funções. Esta seção descreve o processo de seleção, instalação e preparação do ambiente no Raspberry Pi 5, estabelecendo o alicerce sobre o qual todas as funcionalidades de segurança foram implementadas. As etapas aqui detalhadas são cruciais para garantir que o dispositivo opere de forma eficiente, segura e confiável como um *appliance* de segurança de rede.

4.3.1 Seleção e instalação do sistema operacional

A escolha do sistema operacional foi um passo central, guiada pelos critérios de estabilidade, compatibilidade de hardware e suporte da comunidade. Para este projeto, foi selecionado o Raspberry Pi OS (64-bit), uma distribuição baseada em Debian Linux e oficialmente mantida pela Raspberry Pi *Foundation*. A justificativa para esta escolha baseia-se em múltiplos fatores:

- Estabilidade e Confiabilidade: Sendo baseado em Debian, o Raspberry Pi OS herda uma reputação de estabilidade, um requisito indispensável para um dispositivo de segurança que precisa operar continuamente.
- Otimização para o Hardware: Sendo o sistema oficial, ele oferece a melhor compatibilidade e otimização para o hardware do Raspberry Pi 5, garantindo que todos os componentes, como as interfaces de rede e a CPU, funcionem com o máximo de desempenho. A versão de 64-bits foi especificamente escolhida para aproveitar todo o potencial da arquitetura ARMv8 do processador, permitindo melhor gerenciamento de memória e maior poder de processamento, o que é importante para as ferramentas de segurança que demandam mais recursos, como o Suricata.
- Amplo Suporte a Pacotes: O vasto repositório de pacotes do Debian facilita a instalação e a gestão de todas as ferramentas de segurança open-source selecionadas para o projeto, como nftables, Suricata, Squid e WireGuard.

- Comunidade Ativa: Possui uma comunidade global de desenvolvedores e usuários, o que garante acesso a uma vasta documentação, tutoriais e suporte para a resolução de problemas.

A instalação foi realizada utilizando a ferramenta oficial Raspberry Pi Imager, que simplifica o processo de gravação da imagem do sistema no cartão microSD. Optou-se pela versão *headless* (sem ambiente gráfico), uma decisão deliberada para minimizar o consumo de recursos de hardware (CPU e RAM) e, de forma mais importante, para reduzir a superfície de ataque do sistema. A ausência de uma interface gráfica e de serviços associados significa menos software em execução, menos portas abertas e, consequentemente, menos vulnerabilidades potenciais a serem exploradas.

4.3.2 Configurações iniciais do sistema

Após a primeira inicialização, uma série de configurações iniciais foi aplicada para preparar o ambiente para seu papel como *appliance* de *firewall.* A primeira etapa consistiu em habilitar o acesso remoto seguro via *Secure Shell* (SSH). Essa medida é para permitir a administração do dispositivo de forma remota e segura a partir da rede interna, eliminando a necessidade de acesso físico constante e garantindo que a gestão do *firewall* seja realizada apenas por meio de um canal criptografado e a partir de redes confiáveis.

Em seguida, aplicando o princípio da funcionalidade mínima, foram desativados todos os serviços considerados desnecessários para a função de *firewall*. Foram desabilitados, por exemplo, os serviços de Bluetooth e de impressão. Manter apenas os processos estritamente essenciais em execução contribui diretamente para a segurança, ao reduzir a complexidade e a superfície de ataque do sistema, e também otimiza o uso dos recursos limitados de hardware do Raspberry Pi.

4.3.3 Hardening básico e atualizações

O processo de *hardening* (enrijecimento) do sistema operacional é uma prática de segurança essencial que visa tornar o sistema mais resistente a ataques,

minimizando sua superfície de vulnerabilidade. Antes de instalar os serviços de *firewall*, um *hardening* básico foi realizado, focando em duas ações primordiais.

A primeira medida foi a alteração imediata das credenciais padrão de usuário do sistema. Dispositivos como o Raspberry Pi são distribuídos com nomes de usuário e senhas conhecidos, que representam um risco de segurança crítico se não forem alterados. A manutenção de credenciais padrão é uma das falhas mais comuns e exploradas por atacantes para obter acesso não autorizado.

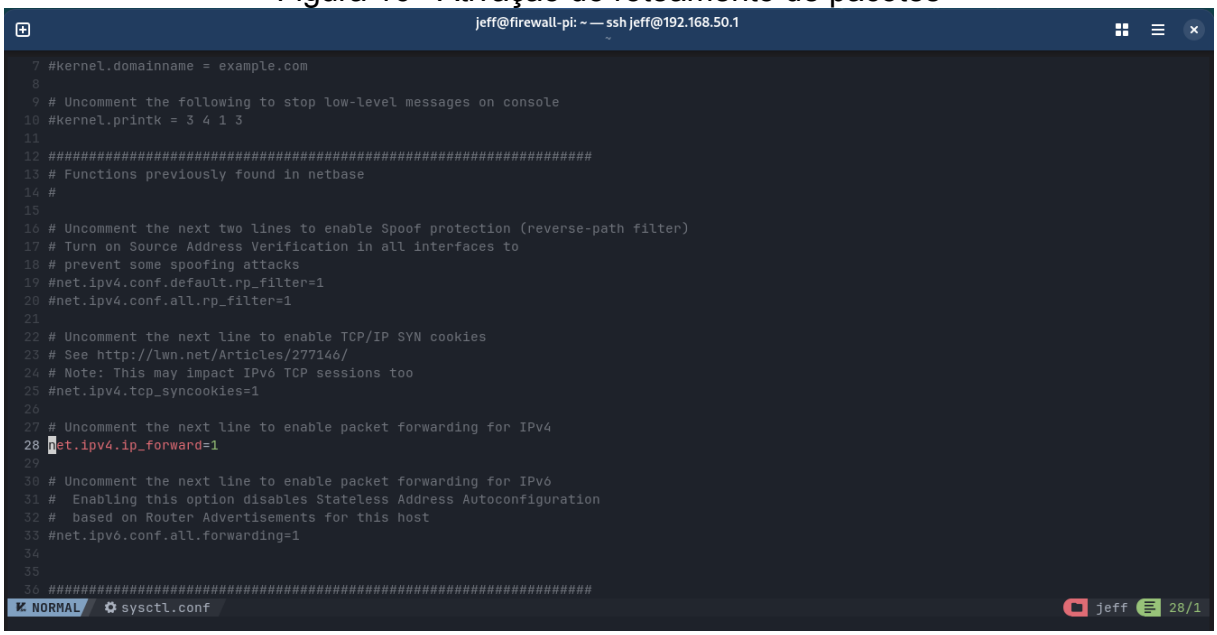
A segunda e mais importante etapa foi a atualização completa do sistema. Imediatamente após a instalação, incluindo o kernel Linux e bibliotecas essenciais, estivessem em suas versões mais recentes. Este procedimento corrige quaisquer vulnerabilidades de segurança já conhecidas e publicadas pelos desenvolvedores. Essa prática de manter o sistema atualizado é um procedimento contínuo e primordial para a manutenção da segurança ao longo de todo o ciclo de vida do firewall.

Adicionalmente, foram aplicadas configurações de limites de recursos (ulimit) para prevenir ataques de exaustão de recursos (DoS) que pudessem travar o kernel via fork bombs. Também optou-se pela desativação do protocolo IPv6 em nível de kernel nas interfaces onde este não é estritamente necessário, reduzindo a superfície de ataque e mitigando vetores de tunelamento não monitorados que poderiam evadir as regras de firewall IPv4.

4.3.4 Configuração de rede base

Para que o Raspberry Pi pudesse atuar como um *gateway* de rede, intermediando o tráfego entre a rede interna e a internet, foi necessário realizar uma configuração de rede base. A principal alteração consistiu na ativação do roteamento de pacotes no kernel Linux. Essa funcionalidade, desativada por padrão, foi habilitada alterando o parâmetro “net.ipv4.ip_forward” para “1” no arquivo de configuração “sysctl.conf” conforme ilustrado na Figura 16, permitindo que o sistema operacional encaminhe pacotes entre diferentes interfaces e transforme, efetivamente, o dispositivo em um roteador.

Figura 16 - Ativação do roteamento de pacotes



Fonte: Autoria Própria

As interfaces de rede físicas foram configuradas conforme a topologia definida no projeto:

- Interface WAN (eth0): Manteve a configuração padrão de cliente DHCP, recebendo um endereço IP automaticamente do roteador de borda que simula a internet.
- Interface LAN (eth1): Recebeu um endereço IP estático (192.168.50.1). A atribuição de um IP fixo é mandatória para um *gateway* de rede, pois ele serve como o ponto de referência para todos os dispositivos da rede interna.

Com o roteamento ativado e as interfaces devidamente configuradas, o Raspberry Pi tornou-se funcional como um *gateway* de rede. Esta configuração de base estabeleceu a fundação necessária para a implementação das segmentações de rede (DMZ, Guest) e para a aplicação das regras de *firewall* detalhadas que serão discutidas na seção seguinte. Para complementar essa configuração e tornar a rede de visitantes (Guest) operacional, foi implementado um servidor DHCP. A ferramenta escolhida foi o dnsmasq, um software leve e versátil que, além de servidor DHCP, também atua como cache de DNS, sendo ideal para dispositivos com recursos limitados. Sua função neste projeto é prover, de forma automática e segura, a configuração de rede para os dispositivos que se conectam à rede Wi-Fi de visitantes. A configuração aplicada pode ser visualizada na Figura 17.

Figura 17 - Configuração do dnsmasq para a rede de visitantes

```
# Configuração WiFi Guest
interface=wlan0
bind-dynamic

# DNS options
domain-needed
bogus-priv

# DHCP para rede Guest (192.168.30.0/24)
dhcp-range=interface:wlan0,192.168.30.10,192.168.30.100,255.255.255.0,12h

# Gateway e DNS para clientes Guest
dhcp-option=interface:wlan0,option:router,192.168.30.1
dhcp-option=interface:wlan0,option:dns-server,192.168.30.1,8.8.8.8

# Não fornecer DHCP em outras interfaces
no-dhcp-interface=eth0,eth1,eth1.20,eth1.40
```

Fonte: Autoria Própria

A configuração do dnsmasq, exibida na Figura 14, foi elaborada para garantir o isolamento da rede de visitantes. A diretiva interface=wlan0 instrui o serviço a operar exclusivamente na interface de rede sem fio. A linha dhcp-range define o escopo de endereços IP que serão distribuídos (de 199.168.30.10 a 192.168.30.100), enquanto a dhcp-option informa aos clientes qual é o gateway (o próprio Raspberry Pi, com IP 192.168.30.1) e quais servidores DNS utilizar. A diretiva de segurança mais importante é a no-dhcp-interface, que proíbe explicitamente o dnsmasq de operar em qualquer outra interface, assegurando que o serviço de DHCP não seja acidentalmente oferecido às redes corporativas e mantendo a segmentação de rede intacta.

4.4 Implementação das Funcionalidades de Firewall

Com a infraestrutura de base e o sistema operacional devidamente preparados, a etapa seguinte consistiu na implementação e configuração do núcleo de segurança da solução. Esta seção detalha a instalação e o ajuste das ferramentas selecionadas para prover as funcionalidades de firewall, detecção de intrusão e filtragem de conteúdo web, traduzindo os requisitos da política de segurança em mecanismos técnicos operacionais.

4.4.1 Motor de Firewall

O componente central da arquitetura de segurança é o motor de firewall, responsável pela filtragem de pacotes, controle de tráfego e tradução de endereços de rede (NAT). Para esta função, foi escolhido o nftables.

A seleção do nftables como motor de firewall em detrimento de seu predecessor, o iptables, foi uma decisão técnica baseada em vantagens de desempenho, sintaxe e flexibilidade. O nftables é o framework de filtragem de pacotes moderno e integrado ao kernel Linux, projetado para superar as limitações do iptables. Suas principais vantagens incluem:

- Desempenho Superior: A arquitetura do nftables permite um processamento de regras mais eficiente, resultando em menor latência e maior *throughput*, fatores críticos em um dispositivo com recursos de hardware limitados como o Raspberry Pi.
- Sintaxe Simplificada e Unificada: Oferece uma sintaxe mais coesa e legível, que permite gerenciar regras de IPv4, IPv6 e outras famílias de protocolos em um único utilitário, simplificando a administração.
- Estrutura Flexível: A utilização de tabelas e cadeias (*chains*) personalizáveis permite uma organização mais lógica e granular das regras, facilitando a implementação de políticas de segurança complexas, como a segmentação de rede.
- Integração Nativa: Possui melhor integração com outras ferramentas do ecossistema Linux, como o Suricata, através de mecanismos como o NFQUEUE.

Essas características tornam o nftables a escolha ideal para uma solução de firewall que busca ser ao mesmo tempo ser confiável, performática e de fácil manutenção.

A arquitetura do nftables foi configurada para atuar nos *hooks* padrão do Netfilter. O tráfego é primeiramente analisado na cadeia de prerouting para decisões de DNAT, segue para a cadeia de forward onde a política de segurança e segmentação é aplicada, e finalmente passa pelo postrouting para aplicação de SNAT (Masquerade) na interface de saída. A utilização de mapas e conjuntos (*sets*) nativos do nftables permitiu que múltiplas regras de verificação de IP fossem

condensadas em uma única operação de busca O(1), otimizando o uso de ciclos de CPU durante a filtragem de pacotes.

A instalação do nftables foi realizada através do gerenciador de pacotes do Raspberry Pi OS. Após a instalação, a configuração inicial envolveu a criação da estrutura para a política de segurança: a definição de tabelas e cadeias. Foram criadas tabelas para filtragem (*filter*) e para tradução de endereços (nat), contendo as cadeias padrão de input (tráfego destinado ao próprio firewall), output (tráfego originado pelo firewall) e *forward* (tráfego que atravessa o firewall, sendo roteado entre interfaces).

A política padrão (*default policy*) para as cadeias input e forward foi definida como drop (descartar). Esta abordagem, conhecida como "negação por padrão" (*default deny*), é uma prática de segurança fundamental que garante que apenas o tráfego explicitamente permitido pelas regras seja autorizado, bloqueando todo o resto.

Com a estrutura inicial estabelecida, foram implementadas as regras de firewall para atender aos requisitos da "Conecta Soluções". As políticas foram construídas com base nos seguintes princípios:

- Inspeção de Estado (*Stateful Inspection*): A primeira regra em todas as cadeias de tráfego de entrada e encaminhamento foi configurada para aceitar pacotes pertencentes a conexões já estabelecidas e relacionadas (ct state established,related accept). Isso otimiza o desempenho, pois o firewall não precisa reavaliar cada pacote de uma comunicação legítima já iniciada.
- Tradução de Endereços de Rede (NAT): Foi implementada uma regra de masquerade na cadeia de postrouting da tabela nat. Essa regra permite que todos os dispositivos das redes internas (LAN, DMZ, Guest, Servidores Privados e VPN) compartilhem o único endereço IP público da interface WAN para acessar a internet, mascarando seus IPs internos.
- Segmentação de Rede: Foram criadas regras explícitas na cadeia forward para controlar o fluxo de tráfego entre as diferentes zonas de segurança. Por exemplo, foi permitido o tráfego da LAN para a WAN e para a DMZ, mas bloqueado o tráfego da rede de visitantes (Guest) para qualquer outra rede interna, garantindo seu completo isolamento.

- Controle de Acesso a Serviços: Regras específicas foram criadas na cadeia input para permitir o acesso a serviços hospedados no próprio firewall, como o acesso SSH (porta 22) apenas a partir da LAN e da VPN, e a porta do servidor WireGuard (UDP 51820) a partir da WAN.

A implementação dessas regras estabeleceu uma base de segurança sólida, controlando o fluxo de dados e protegendo o perímetro da rede corporativa simulada.

4.4.2 Sistema de Detecção de Intrusão

Para complementar a filtragem de pacotes do nftables com uma camada de segurança mais profunda, foi implementado um Sistema de Detecção e Prevenção de Intrusão (IDS/IPS).

Enquanto o nftables opera primariamente nas camadas 3 e 4 do modelo OSI (IP e portas), um *firewall* moderno precisa de visibilidade sobre o conteúdo das comunicações (camada 7) para identificar ameaças sofisticadas. É neste ponto que o Suricata diferencia a solução proposta de um *firewall* comum, elevando o Raspberry Pi à categoria de um NGFW. Ao analisar a Camada 7, o Suricata verifica não apenas quem está conectando, mas o que está sendo enviado dentro da conexão. A política de segurança da empresa exige a prevenção ativa contra ataques como *malware*, *phishing* e varreduras de portas. O Suricata foi escolhido por ser uma solução IDS/IPS de alto desempenho que realiza Inspeção Profunda de Pacotes (DPI), permitindo a detecção de ameaças com base em assinaturas e anomalias de protocolo, funcionalidades essenciais para um NGFW.

A integração entre o firewall e o IPS ocorre através do mecanismo NFQUEUE (Netfilter Queue). Diferente de um IDS passivo que apenas recebe uma cópia do tráfego, o modo IPS inline configurado intercepta o pacote na cadeia de forward do kernel e o envia para o espaço de usuário (*userspace*), onde o Suricata realiza a inspeção profunda. O pacote permanece retido no kernel até que o Suricata emita um veredicto (ACCEPT ou DROP). Para garantir a segurança da rede em caso de falha do processo do Suricata, a fila foi configurada em modo fail-closed (falha fechada), garantindo que, se o serviço de inspeção parar, o tráfego seja interrompido em vez de passar sem análise.

O Suricata foi instalado a partir dos repositórios do sistema operacional. A configuração principal envolveu a definição das interfaces de rede a serem monitoradas e o gerenciamento dos conjuntos de regras. Foi habilitado o conjunto de regras gratuito ET Open (Emerging Threats), que fornece uma base ampla e atualizada de assinaturas para ameaças conhecidas. As regras foram ajustadas para se adequarem ao perfil da empresa, habilitando categorias relevantes como detecção de *malware*, tráfego de *command & control* (C&C) e tentativas de exploração de vulnerabilidades.

Para que o Suricata pudesse atuar como um Sistema de Prevenção de Intrusão (IPS), capaz de bloquear ativamente o tráfego malicioso, foi necessário integrá-lo ao nftables. Essa integração foi realizada configurando o Suricata para operar em modo *inline*, utilizando a fila de pacotes do kernel Linux (NFQUEUE).

Uma regra específica foi adicionada ao nftables na cadeia forward, instruindo o firewall a enviar todo o tráfego que atravessa o gateway para a fila de número 0 (queue num 0). O Suricata, por sua vez, foi configurado para "escutar" essa fila, inspecionar cada pacote recebido e, com base em suas regras, decidir se o pacote deve ser aceito (accept), descartado (drop) ou modificado. Esta arquitetura permite que o Suricata inspecione o tráfego em tempo real antes que ele alcance seu destino, bloqueando ameaças de forma proativa.

4.4.3 Proxy Web e Filtragem de Conteúdo

Para atender aos requisitos de controle de uso da internet e bloqueio de conteúdo inadequado, foi implementada uma solução de *proxy* web.

A política de segurança da "Conecta Soluções" estipula que o acesso à internet é uma ferramenta de trabalho e, portanto, deve ser controlado para mitigar riscos de segurança e garantir a produtividade. A implementação de um proxy com capacidade de filtragem de conteúdo é necessária para:

- Bloquear o acesso a sites que representam riscos de segurança (malware, phishing).
- Restringir o acesso a categorias de conteúdo inadequadas para o ambiente de trabalho (conteúdo adulto, apostas).

- Controlar o acesso a serviços que consomem muita banda, como redes sociais e streaming, durante o horário de expediente.
- Registrar (log) toda a atividade de navegação para fins de auditoria e conformidade.

A solução implementada utiliza o Squid como servidor proxy, configurado para operar em modo transparente. A principal vantagem deste modo é que ele intercepta todo o tráfego web (portas 80 e 443) sem a necessidade de configurar manualmente cada estação de trabalho cliente.

Para viabilizar a operação transparente, foi criada uma regra de redirecionamento no nftables. Essa regra intercepta todos os pacotes TCP destinados à porta 80 (HTTP) originados da LAN e da rede de visitantes, e os redireciona para a porta local onde o serviço do Squid está em execução (porta 3129). Dessa forma, todo o tráfego de navegação é forçado a passar pelo proxy para inspeção.

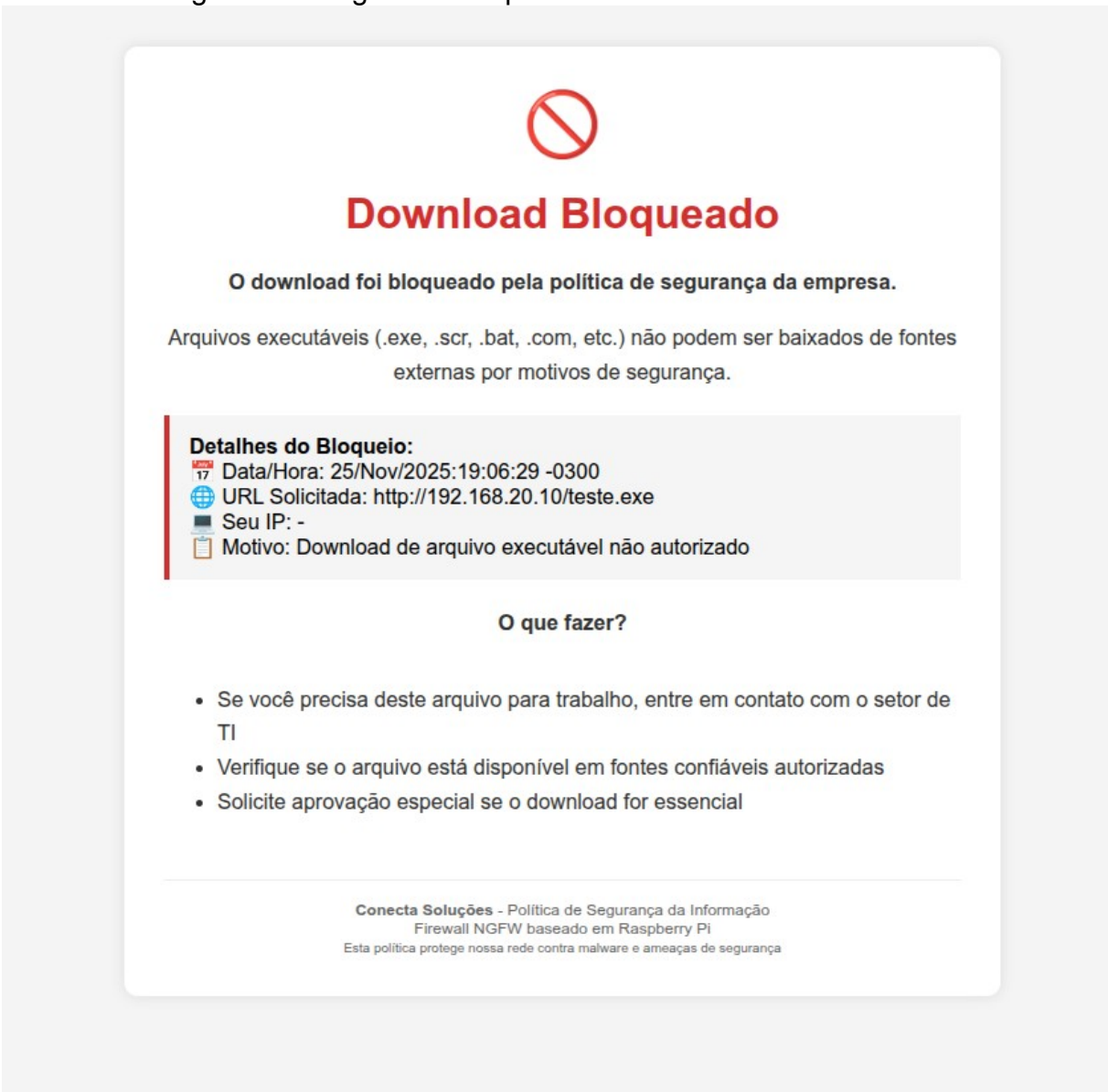
Para realizar a filtragem de conteúdo baseada em categorias, o Squid foi integrado ao SquidGuard, um redirecionador de URLs flexível. O SquidGuard foi configurado para utilizar listas de bloqueio (*blacklists*) públicas e gratuitas, que contêm milhões de domínios categorizados.

Foram criadas Listas de Controle de Acesso (ACLs) para implementar as políticas da empresa, incluindo:

- Bloqueio permanente de categorias como malware, phishing, porn e gambling.
- Bloqueio por horário para as categorias social_media e streaming, restringindo o acesso durante o horário de expediente (08:00 às 18:00) e liberando-o fora desse período.
- Bloqueio de tipos de arquivo, impedindo o download de arquivos executáveis (extensões .exe, .scr, .bat) para mitigar a entrada de *malware*.

A efetividade das Listas de Controle de Acesso (ACLs) pode ser observada na intercepção de downloads perigosos. Ao tentar baixar um arquivo executável, o tráfego é interrompido e o usuário é redirecionado para uma página informativa contendo os detalhes do bloqueio e a política infringida, como evidenciado na Figura 18.

Figura 18 - Página de bloqueio de download não autorizado



Fonte: Autoria Própria

4.5 Funcionalidades Avançadas

Além das funcionalidades essenciais de filtragem de pacotes, um firewall de próxima geração (NGFW) deve prover capacidades avançadas que enderecem as necessidades modernas de segurança e conectividade corporativa. Para a solução implementada, duas áreas foram consideradas críticas: a garantia de acesso remoto seguro para colaboradores e a implementação de um sistema robusto de monitoramento e registro de eventos. Estas funcionalidades não apenas fortalecem a postura de segurança da organização, mas também fornecem a visibilidade

necessária para auditoria, conformidade com regulamentações como a LGPD e resposta a incidentes.

4.5.1 VPN

O uso de uma Rede Privada Virtual (VPN) é uma necessidade para empresas como a "Conecta Soluções", que operam em modelos de trabalho flexíveis, incluindo o formato híbrido. A VPN estabelece um canal de comunicação seguro e criptografado sobre redes públicas, como a internet, permitindo que usuários remotos acessem recursos da rede interna como se estivessem fisicamente presentes no escritório.

Conforme a política de segurança e o perfil operacional da empresa fictícia, cerca de 20% dos seus colaboradores necessitam de acesso regular a sistemas internos, como servidores de arquivos e bancos de dados, a partir de locais externos. A política de segurança da informação estipula que todo acesso a recursos internos a partir da internet deve ocorrer exclusivamente por meio de um canal seguro e criptografado, em conformidade com o princípio do menor privilégio.

Dessa forma, a implementação de um servidor VPN no próprio *firewall* centraliza o ponto de entrada, simplifica a gestão de acessos e garante que todo o tráfego remoto seja inspecionado pelas mesmas políticas de segurança aplicadas ao tráfego interno.

Para esta funcionalidade, foi selecionada a ferramenta WireGuard. A escolha foi motivada por sua arquitetura moderna, simplicidade de configuração, alto desempenho e uso de criptografia de ponta, características que a tornam particularmente adequada para hardware com recursos limitados como o Raspberry Pi, sem comprometer a segurança.

A configuração do servidor no Raspberry Pi envolveu os seguintes passos:

1. Instalação e Geração de Chaves: O pacote do WireGuard foi instalado a partir dos repositórios do sistema operacional. Em seguida, foi gerado um par de chaves criptográficas (pública e privada) para o servidor. Cada cliente (colaborador remoto) também deve possuir seu próprio par de chaves.
2. Criação da Interface Virtual: Foi criada uma interface de rede virtual, denominada wg0, que opera como o ponto de terminação do túnel VPN no

firewall. A esta interface foi atribuído o endereço IP 10.10.10.1, servindo como gateway para a sub-rede da VPN (10.10.10.0/24).

3. Configuração do Servidor: O arquivo de configuração do servidor foi criado para escutar por conexões na porta UDP 51820. Para cada cliente autorizado, uma seção [Peer] foi adicionada, contendo sua chave pública e o endereço IP que lhe será atribuído dentro da rede VPN.

A configuração do servidor VPN foi realizada de forma declarativa, garantindo persistência e segurança. A Figura 19 exibe o conteúdo do arquivo de configuração da interface, definindo a porta de escuta (UDP 51820), o endereçamento do túnel privado e a associação das chaves públicas dos clientes autorizados (*Peers*).

Figura 19 - Arquivo de configuração da interface WireGuard

```
firewall-pi# pwd
/etc/wireguard
firewall-pi# ls
wg0.conf
firewall-pi# cat wg0.conf
[Interface]
PrivateKey = QP219+f+d1QUonhJAAUyXF/T70vJjbNIASvPT7WaKmc=
Address = 10.10.10.1/24
ListenPort = 51820

[Peer]
PublicKey = Yy3krX7oDEs17AH3djm70R1f6RPNnjdN70aReuge1i0=
AllowedIPs = 10.10.10.2/32
firewall-pi#
```

Fonte: Autoria Própria

A integração da VPN com o firewall foi finalizada através da definição de regras específicas no nftables, garantindo que o tráfego dos usuários remotos fosse devidamente controlado:

- Permissão de Acesso ao Servidor: Uma regra foi adicionada à cadeia input para permitir a entrada de pacotes UDP na porta 51820 vindos da interface WAN, possibilitando que os clientes estabeleçam a conexão VPN.
- Roteamento do Tráfego: Regras na cadeia forward foram implementadas para permitir que o tráfego originado da rede VPN (VPN_NET) seja

encaminhado para as redes internas de destino, especificamente a LAN Corporativa (LAN_NET) e a DMZ (DMZ_NET). O acesso a outras zonas, como a rede de servidores privados, permanece bloqueado por padrão, seguindo a política de segurança.

- Acesso à Internet: Uma regra de masquerade na tabela nat foi aplicada ao tráfego da VPN, permitindo que os clientes remotos também possam utilizar a conexão de internet da empresa de forma segura, caso optem por rotear todo o seu tráfego através do túnel.

4.5.2 Monitoramento e Logs

Controlar e analisar os eventos gerados na rede é parte importante da segurança da informação. Na "Conecta Soluções", esse acompanhamento constante é obrigatório para atender às exigências da LGPD, facilitando a auditoria de acessos e a identificação rápida de ações suspeitas.

Para atender a este requisito de forma eficiente em um hardware com recursos limitados, foi implementada a pilha de ferramentas Promtail, Loki e Grafana. Esta abordagem, conhecida como "PLG", é mais leve e moderna que alternativas como a pilha ELK.

A escolha da pilha PLG (Promtail, Loki, Grafana) em detrimento de soluções tradicionais como ELK (Elasticsearch, Logstash, Kibana) é tecnicamente justificada pelas restrições de hardware. Enquanto o Elasticsearch indexa o conteúdo completo dos logs (Full Text Search), exigindo grande quantidade de memória RAM e I/O de disco, o Loki adota uma abordagem de indexação baseada apenas em metadados (labels), comprimindo os chunks de logs brutos. Isso reduz drasticamente o consumo de memória e armazenamento, viabilizando uma solução de monitoramento centralizada rodando no próprio Raspberry Pi sem competir excessivamente por recursos com os serviços de segurança.

A arquitetura funciona da seguinte forma:

- Promtail: Atua como o agente de coleta, instalado diretamente no Raspberry Pi. Ele foi configurado para monitorar os arquivos de log gerados por todos os serviços de segurança (nftables, Suricata, Squid) e pelo próprio sistema (syslog).

- Loki: É o sistema de agregação de logs, responsável por receber os dados enviados pelo Promtail, indexá-los com base em metadados (como serviço de origem e nível de severidade) e armazená-los de forma otimizada para consulta.
- Grafana: Funciona como a interface de visualização e análise. Conectado ao Loki como fonte de dados, o Grafana permite a criação de painéis (dashboards) interativos para explorar os logs de maneira intuitiva.

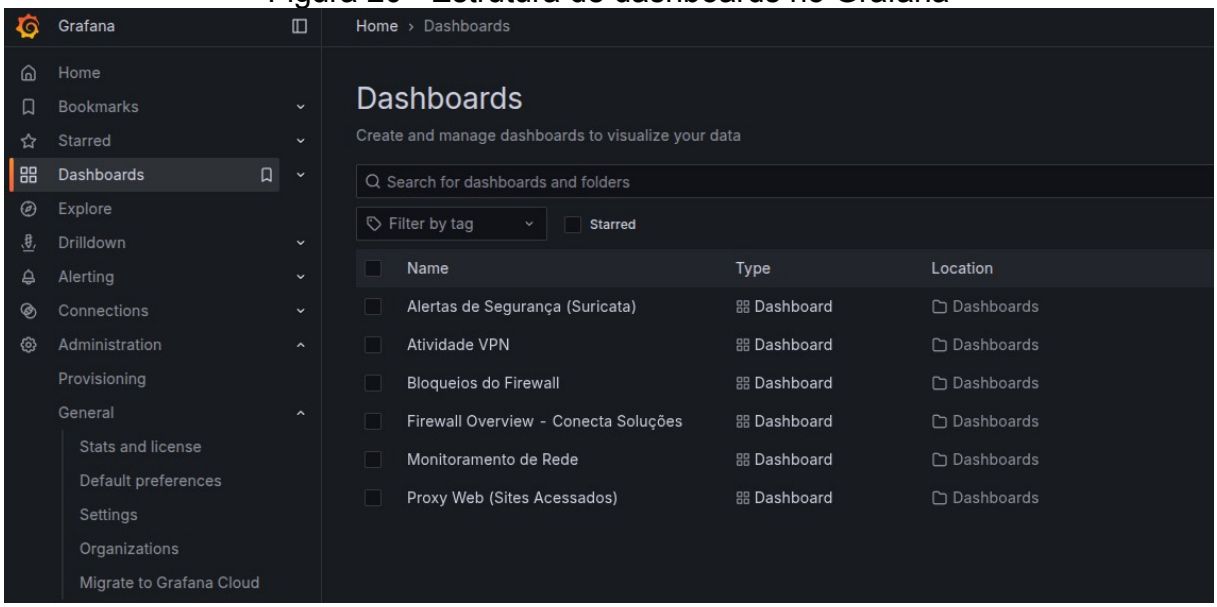
Duas ferramentas principais foram configuradas para prover visibilidade sobre a atividade da rede:

1. Grafana: Foram desenvolvidos dashboards específicos para visualizar métricas críticas de segurança em tempo real, como:
   - Tentativas de conexão bloqueadas pelo nftables, categorizadas por origem e destino.
   - Alertas de segurança de alta prioridade gerados pelo Suricata.
   - Os sites mais acessados e bloqueados pelo Squid, permitindo a análise do uso da internet.
   - Atividade de conexão dos usuários da VPN, incluindo logins e volume de tráfego.
2. ntopng: Para uma análise de tráfego em tempo real e de alto nível, foi instalado o ntopng. Esta ferramenta oferece uma interface web que exibe os fluxos de rede ativos, os principais consumidores de banda ("top talkers"), os protocolos mais utilizados e outras estatísticas vitais, permitindo ao time de TI identificar rapidamente anomalias ou gargalos de desempenho na rede.

Para garantir uma resposta proativa a ameaças, o sistema de monitoramento foi configurado para ir além da simples visualização. Utilizando o sistema de alertas nativo do Grafana, foram criadas regras para notificar a equipe de TI sobre eventos críticos. Por exemplo, foram configurados alertas para serem disparados caso o número de tentativas de acesso bloqueadas em um curto intervalo de tempo exceda um limiar predefinido, ou quando um alerta de severidade crítica for gerado pelo Suricata. Essas notificações podem ser integradas a canais como e-mail ou sistemas de mensageria, permitindo uma ação rápida para investigar e mitigar potenciais incidentes de segurança.

A organização dos dados coletados foi estruturada em painéis temáticos, facilitando a correlação de eventos e a auditoria do sistema. Conforme demonstrado na Figura 20, foram criados dashboards específicos para cada componente crítico, abrangendo desde alertas do Suricata até o monitoramento de atividade da VPN.

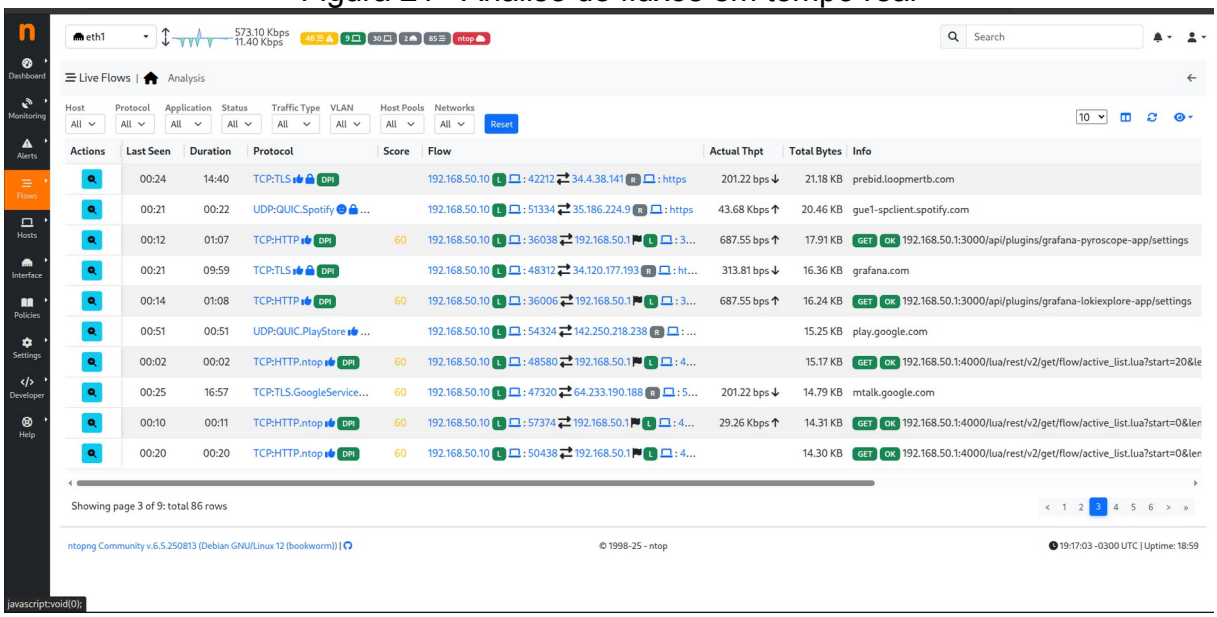
Figura 20 - Estrutura de dashboards no Grafana



Fonte: Autoria Própria

A visibilidade granular do tráfego é garantida pela inspeção de fluxos em tempo real. A Figura 21 demonstra a capacidade da ferramenta ntopng em identificar e classificar protocolos de camada de aplicação (L7), como fluxos de Spotify e Google Services, permitindo uma auditoria detalhada do uso da banda. Para uma visão macroscópica da rede, utilizam-se painéis gerenciais que consolidam as métricas de consumo.

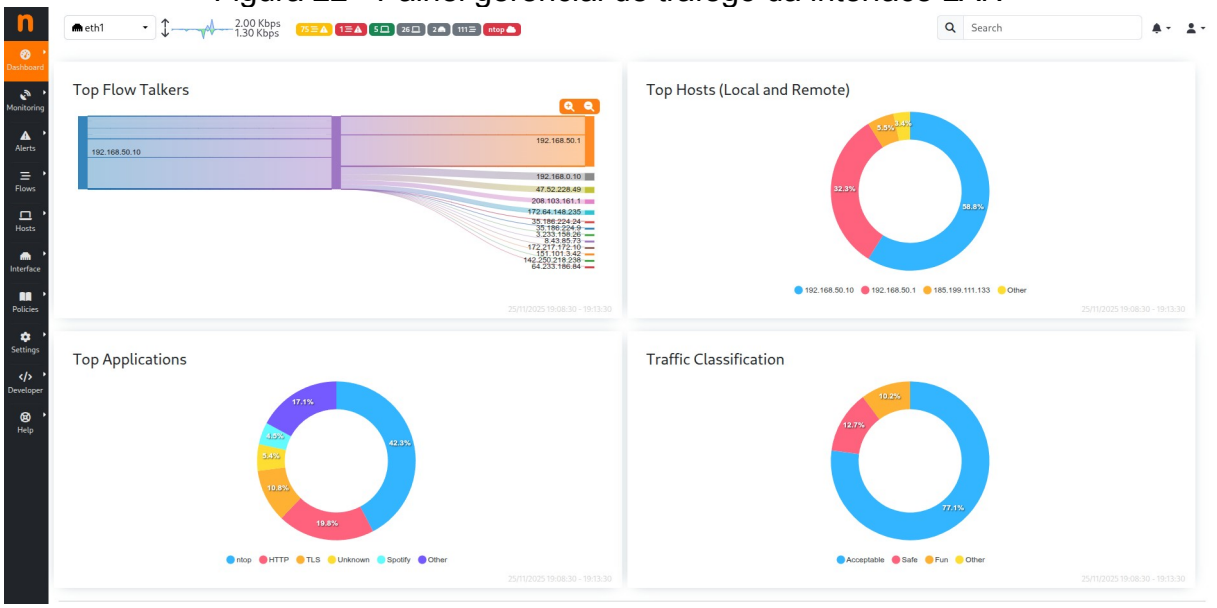
Figura 21 - Análise de fluxos em tempo real



Fonte: Autoria Própria

A Figura 22 ilustra o dashboard da interface interna, destacando os hosts com maior volume de tráfego e a categorização das aplicações em uso na organização.

Figura 22 - Painel gerencial de tráfego da interface LAN

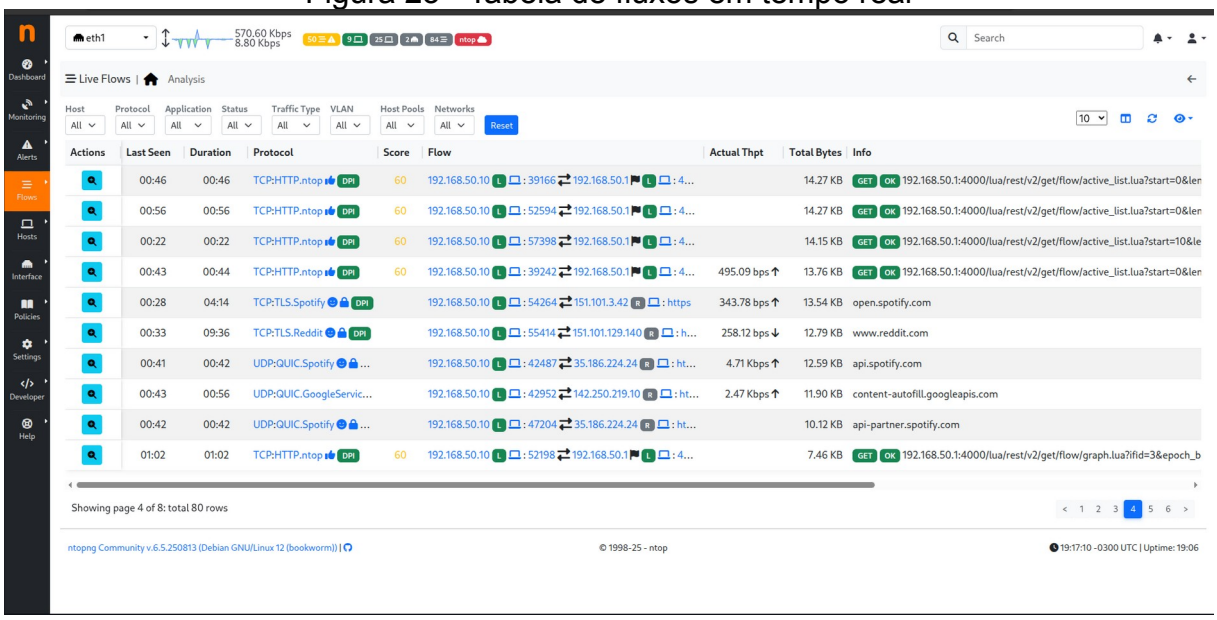


Fonte: Autoria Própria

A capacidade de *Deep Packet Inspection* (DPI) passiva da ferramenta ntopng permite identificar o uso de aplicações específicas, independentemente da porta utilizada. A Figura 23 evidencia a detecção precisa de fluxos de redes sociais

(Reddit) e streaming de áudio (Spotify) originados de um host da LAN, validando a visibilidade de camada 7 (aplicação) sobre o tráfego criptografado.

Figura 23 - Tabela de fluxos em tempo real

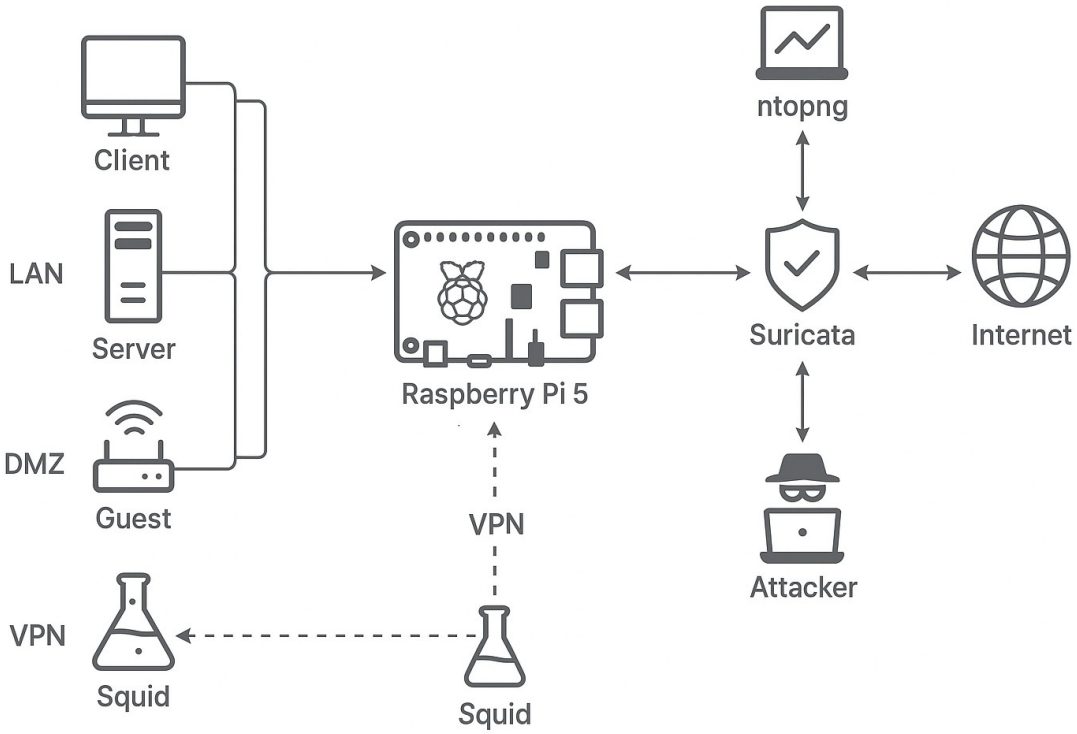


Fonte: Autoria Própria

5 PLANEJAMENTO E EXECUÇÃO DOS TESTES

Com o ambiente de firewall implementado e configurado conforme descrito no capítulo 4, a etapa seguinte consiste na realização de testes práticos para avaliar de forma sistemática a viabilidade técnica do Raspberry Pi 5 como solução de firewall de baixo custo. Os testes foram planejados para reproduzir cenários reais enfrentados por pequenas e médias empresas, contemplando métricas de desempenho, estabilidade, consumo de recursos e segurança, alinhadas aos objetivos centrais deste trabalho. A Figura 24 apresenta a topologia geral do ambiente de testes utilizado neste trabalho, ilustrando o Raspberry Pi 5 atuando como firewall principal, interligando as redes LAN, DMZ, Guest, WAN e VPN, bem como as principais ferramentas empregadas para inspeção, filtragem e monitoramento.

Figura 24 - Topologia geral do ambiente de testes e do firewall



Fonte: Autoria Própria

Essa visão global auxilia na compreensão da infraestrutura utilizada durante as fases de teste descritas a seguir. O planejamento foi estruturado em fases

progressivas, abrangendo desde testes básicos de desempenho de rede até avaliações de segurança, consumo de recursos, estabilidade e integração das funcionalidades implementadas. Essa abordagem possibilita não apenas medir o desempenho bruto do dispositivo, mas também compreender como ele se comporta em situações de uso prolongado e sob diferentes condições de tráfego e ataque.

O plano de testes foi dividido em cinco fases principais:

- Fase 1 - Testes de Desempenho de Rede: medições de throughput, latência, conexões simultâneas e perda de pacotes.
- Fase 2 - Testes de Segurança e Resiliência: avaliação frente a ataques de negação de serviço, tráfego malicioso e filtragem web.
- Fase 3 - Testes de Consumo de Recursos: monitoramento de CPU, memória e temperatura do Raspberry Pi durante diferentes cargas.
- Fase 4 - Testes de Estabilidade e Escalabilidade: análise de uptime contínuo, degradação de performance e capacidade de expansão para cenários maiores.
- Fase 5 - Testes Funcionais Integrados: simulação de cenários corporativos reais, com acesso à internet filtrado, uso de VPN, isolamento de redes e auditoria via logs.

Cada fase é composta por diferentes experimentos, que foram planejados com metodologias específicas, ferramentas adequadas e critérios de coleta de métricas para permitir comparações objetivas.

5.1 Testes de Desempenho de Rede

A primeira fase dos experimentos foi dedicada à análise do desempenho de rede do firewall implementado no Raspberry Pi 5. O objetivo central dessa etapa foi mensurar a capacidade do dispositivo em processar e encaminhar pacotes de dados entre as diferentes interfaces de rede configuradas, validando se ele atende aos requisitos mínimos de uma pequena ou média empresa em termos de velocidade, estabilidade e confiabilidade de comunicação.

Nesta fase, foram analisadas métricas de infraestrutura, incluindo *throughput* (taxa de transferência efetiva), latência (RTT), capacidade de gerenciamento de conexões simultâneas e taxa de perda de pacotes. A obtenção desses indicadores é primordial para estabelecer uma linha de base (*baseline*) de desempenho do

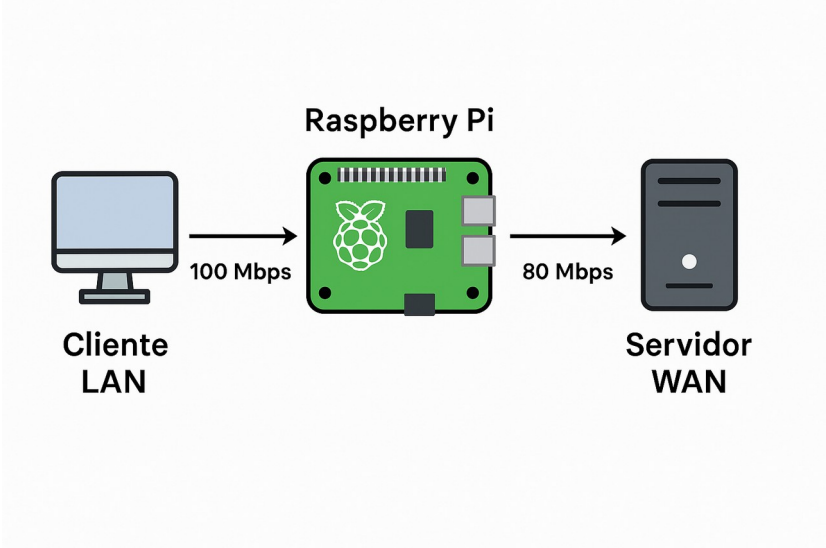
sistema em condições normais e de estresse controlado. Essa caracterização prévia é indispensável para distinguir, nas fases subsequentes, as limitações impostas pelo hardware daquelas decorrentes da ativação de mecanismos de segurança complexos, como o IPS (Sistema de Prevenção de Intrusão).

A metodologia aplicada reproduziu cenários de tráfego sintético e realístico, variando protocolos de transporte (TCP e UDP), tamanhos de carga útil (*payload*) e volume de requisições concorrentes. A coleta sistemática de dados, realizada por meio de ferramentas de monitoramento em tempo real e análise de séries temporais, permitiu a construção de uma visão abrangente sobre o comportamento do firewall sob diferentes perfis de carga.

5.1.1 Testes de Throughput

Este teste avaliou a capacidade máxima de processamento de tráfego do Raspberry Pi atuando como firewall, buscando identificar o *throughput* sustentável sob diferentes cargas de trabalho. A análise foi além da taxa de transferência bruta, observando também a estabilidade da conexão (perda de pacotes e retransmissões) e o impacto no hardware. A topologia do experimento, ilustrada na Figura 25, estabelece a comunicação entre um cliente na LAN e um servidor na WAN, com o firewall interceptando o tráfego.

Figura 25 - Topologia do teste de throughput



Fonte: Autoria Própria

Para a geração e medição do tráfego TCP e UDP, utilizou-se a ferramenta *iperf3*. O monitoramento de recursos do sistema foi realizado via *htop/top*, enquanto o *Grafana* e o *ntopng* permitiram a visualização do desempenho em tempo real.

Cenários executados:

1. Throughput TCP entre LAN e WAN:
    - Configurou-se um servidor *iperf3* em um dispositivo na WAN e um cliente na LAN.
    - Foram realizados testes unidirecionais (LAN→WAN e WAN→LAN) e bidirecionais.
    - Cada teste teve duração mínima de 60 segundos, garantindo estabilização da transmissão TCP.
2. Throughput UDP com diferentes tamanhos de pacotes:
    - Foram executados testes com pacotes de 64, 128, 256, 512, 1024 e 1470 bytes.
    - Em cada configuração, foram medidas a taxa de throughput, perda de pacotes e jitter.
    - Essa etapa permitiu avaliar a eficiência do firewall em cenários de tráfego de alta velocidade com pacotes pequenos e grandes.
3. Throughput com múltiplas conexões simultâneas:
    - Foram abertas conexões TCP em paralelo (1, 5, 10, 25, 50 e 100 streams simultâneos).
    - Durante os testes, monitorou-se o impacto sobre CPU e memória, verificando o comportamento do firewall em cenários de alta concorrência de usuários.

Durante as execuções, foram registrados os valores máximos, médios e mínimos de *throughput*, bem como o desvio padrão para análise de estabilidade. Complementarmente, monitorou-se a temperatura do processador e o uso de recursos do Raspberry Pi, consolidando os dados quantitativos necessários para validar o desempenho do dispositivo como elemento central da rede.

5.1.2 Testes de Latência

O segundo conjunto de experimentos teve como objetivo medir o atraso introduzido pelo firewall implementado no Raspberry Pi 5 durante a transmissão de

pacotes. Esse teste é essencial para avaliar o impacto do dispositivo em aplicações sensíveis a tempo de resposta, como chamadas de voz (VoIP), jogos online ou sistemas corporativos que exigem comunicação em tempo real.

Para a execução, foram utilizadas ferramentas tradicionais de medição de latência, como ping e fping, além de utilitários mais avançados como hping3, netperf e mtr, que possibilitaram testar diferentes protocolos e tamanhos de pacotes. Em cenários específicos, também foram aplicados owamp/twamp para medições unidirecionais mais precisas.

Foram planejados e realizados os seguintes cenários de teste:

1. Latência Base (Baseline):
    - Cliente na LAN enviando pacotes ICMP para um servidor na WAN (Google DNS – 8.8.8.8).
    - Utilizou-se pacotes padrão de 64 bytes, com 1000 pings em intervalos de 1 segundo.
    - Foram registrados RTT mínimo, máximo, médio, jitter e perda de pacotes.
2. Variação do Tamanho de Pacote:
    - Foram enviados pacotes ICMP com diferentes tamanhos (64, 128, 256, 512, 1024 e 1472 bytes).
    - Cada teste contou com 100 pings para cada tamanho de pacote, medindo o efeito do aumento de payload na latência.
3. Latência com Diferentes Protocolos:
    - Além do ICMP tradicional, foram realizados testes com pacotes TCP SYN e UDP usando o hping3.
    - Esse cenário permitiu avaliar o impacto do firewall ao processar protocolos distintos, considerando o funcionamento do DPI (Deep Packet Inspection).
4. Latência Sob Carga:
    - Durante a execução de tráfego gerado com o iperf3 em diferentes intensidades (50 Mbps e 200 Mbps), foram executados testes de ping em paralelo.
    - Essa configuração permitiu observar o aumento da latência quando o firewall opera em condições de alta utilização de banda.
5. Latência Interna (Entre VLANs):

- Foram testadas as comunicações entre diferentes segmentos de rede configurados no firewall: LAN ↔ DMZ, LAN ↔ Guest, DMZ ↔ WAN, Servidores Privados ↔ LAN e VPN ↔ LAN.
- Cada teste consistiu em 100 pacotes ICMP entre os pares de origem e destino, permitindo verificar o impacto das regras de segmentação no tempo de resposta.

Durante os testes, foram registradas métricas de RTT mínimo, máximo e médio, jitter e perda de pacotes. Também houve monitoramento de CPU, memória e temperatura do Raspberry Pi para identificar possíveis relações entre picos de uso e variações de latência.

5.1.3 Testes de Conexões Simultâneas

O terceiro conjunto de experimentos teve como objetivo avaliar a capacidade do firewall implementado no Raspberry Pi 5 de manter múltiplas conexões TCP ativas de forma estável, identificando o ponto de saturação em que ocorre degradação perceptível no desempenho. Esse tipo de análise é fundamental em cenários corporativos, nos quais diversos usuários acessam simultaneamente serviços internos e externos, exigindo que o dispositivo gerencie adequadamente suas tabelas de estado e os recursos do kernel.

Para a execução dos testes, foi configurado um servidor web simples em um contêiner na rede DMZ, simulando um serviço corporativo acessado pelos usuários da LAN. A partir da rede interna, foram utilizadas as ferramentas Apache Bench (ab) e wrk para a geração de conexões HTTP em diferentes intensidades. Adicionalmente, *scripts* com curl em paralelo foram empregados para criar conexões TCP genéricas com serviços externos, possibilitando observar o comportamento do firewall sob alta concorrência.

Durante os experimentos, o monitoramento do uso de CPU, memória e temperatura do Raspberry Pi foi realizado com o auxílio do htop e do Grafana, enquanto o número real de conexões mantidas foi acompanhado por meio do utilitário conntrack-tools.

Foram considerados três cenários principais:

1. Abertura gradual de conexões HTTP:

- Iniciou-se com 100 conexões simultâneas, avançando progressivamente para 500, 1000, 2000, 5000, 10000, 20000 e 40000 conexões.
- Foram observados o tempo médio de resposta, a taxa de erros de conexão e o impacto nos recursos do sistema.

2. Sustentação de conexões de longa duração:
   - Com a ferramenta wrk, conexões persistentes foram mantidas ativas por períodos entre 5 e 10 minutos, simulando aplicações corporativas (ex.: VPN e sistemas internos).
3. Conexões TCP concorrentes genéricas:
   - Foram abertos múltiplos fluxos simultâneos para serviços externos (como google.com) utilizando scripts com curl.
   - Esse cenário permitiu verificar se o firewall mantinha todas as sessões sem quedas ou resets inesperados.

As métricas coletadas incluíram o número máximo de conexões ativas sustentadas, a taxa de sucesso no estabelecimento de sessões, o tempo médio de resposta das requisições HTTP, além do uso de CPU, memória e temperatura do dispositivo durante o pico de carga. Também foi monitorada a utilização da tabela conntrack, comparando o número de entradas ativas com o limite configurado no kernel. Esses dados fornecem a base necessária para determinar se o Raspberry Pi 5 atende ao requisito definido no perfil da empresa simulada, de suportar pelo menos 10.000 conexões simultâneas de forma estável.

5.1.4 Testes de Perda de Pacotes

O quarto conjunto de testes avaliou a capacidade do Raspberry Pi 5 de encaminhar pacotes sem perdas significativas em diferentes condições de tráfego. A ideia foi verificar se o firewall mantém entregas consistentes mesmo sob carga, tanto entre as VLANs (LAN, DMZ e Guest) quanto no caminho LAN-WAN, algo essencial para uso em uma PME. Para isso, foram definidos cenários que variam intensidade, tamanho de pacotes, concorrência entre VLANs e duração.

A geração e análise do tráfego foram feitas com iperf3 (UDP e TCP), apoiadas por mtr e ping/fping para acompanhar RTT e jitter ao longo do teste. A telemetria de perda, throughput e uso de recursos ficou a cargo do Grafana e do ntopng. A

topologia e as regras do firewall foram as mesmas adotadas nos testes anteriores: política *default deny*, inspeção *stateful*, NAT, segmentação entre as três VLANs, proxy transparente e Suricata operando como IPS via NFQUEUE. Assim, as medições refletem o comportamento real do sistema com todos os mecanismos ativos.

Foram executados quatro cenários, todos com duração mínima de 60 s, salvo o cenário de sustentação, em que o tráfego foi mantido por 10 a 15 minutos:

1. Perda sob diferentes intensidades de tráfego (UDP)
    - Foram injetadas taxas progressivas (50, 100, 200 e 400 Mbps, até o limite suportado), em ambas as direções (LAN→WAN e WAN→LAN). Para cada taxa, registrou-se a perda (%) reportada pelo iperf3, além de jitter e throughput efetivo em relação ao configurado. Em paralelo, realizou-se ping amostrado para coletar RTT médio/máximo e estimar jitter por variação inter-amostra.
2. Perda por tamanho de pacote (UDP)
    - Com a taxa fixa (100 Mbps) e tamanhos de payload de 64, 128, 512, 1024 e 1470 bytes, mediu-se o impacto do PPS (packets per second) na perda. A hipótese operativa, coerente com a literatura, é de que pacotes menores elevam o PPS e tendem a expor limites de processamento primeiro.
3. Perda com tráfego concorrente entre VLANs
    - Para refletir o cenário corporativo segmentado, gerou-se tráfego paralelo originado de LAN, DMZ e Guest (ex.: LAN→WAN UDP 200 Mbps; DMZ→WAN TCP 100 Mbps; Guest→WAN UDP 50 Mbps). Esse arranjo observa se a perda se distribui entre as zonas ou se concentra quando múltiplas filas disputam CPU, buffers e caminho de saída.
4. Sustentação sob carga (longa duração)
    - Mantiveram-se fluxos UDP/TCP por 10–15 minutos para verificar estabilidade temporal da perda. Observou-se se há tendência de aumento (indicando saturação de buffers ou thermal throttling) e se o throughput efetivo decai ao longo do tempo.

Em todos os cenários, o iperf3 foi executado com saída detalhada, enquanto mtr/ping rodaram em paralelo para coleta de métricas auxiliares. Os dashboards registraram CPU, RAM e temperatura do Raspberry Pi, permitindo correlacionar

aumentos de perda com possíveis gargalos. As métricas principais coletadas foram: perda (%), jitter (ms) e throughput efetivo (Mbps). As secundárias incluíram RTT médio/máximo, uso de CPU, RAM e temperatura. Esses registros servem como base para o capítulo seguinte, permitindo identificar o ponto de saturação do dispositivo, avaliar o efeito do PPS e verificar se o tráfego concorrente entre VLANs amplia a perda de forma uniforme ou não.

5.2 Testes de Segurança e Resiliência

Nesta fase, foram realizados os testes voltados à avaliação da segurança e capacidade de resiliência do firewall implementado no Raspberry Pi 5, considerando situações reais de ameaça cibernética e comportamento sob condições de estresse. O objetivo principal dessa etapa foi verificar se o sistema, além de prover filtragem e desempenho satisfatórios, consegue detectar, mitigar e resistir a ataques e tráfegos maliciosos de forma comparável a soluções corporativas.

Os experimentos dessa fase buscaram simular ataques e incidentes típicos em ambientes empresariais, como tentativas de negação de serviço, escaneamentos de rede, injeções de tráfego anômalo e abusos de protocolo. Durante os testes, foram utilizados ferramentas especializadas de geração de ataques e monitoramento, de modo a reproduzir cenários próximos aos que um firewall comercial enfrentaria no dia a dia.

As métricas observadas incluíram uso de recursos (CPU, memória e temperatura), disponibilidade dos serviços, número de conexões simultâneas, efetividade das regras e alertas do IDS/IPS e o tempo de recuperação do sistema após o término dos ataques. Todos os dados coletados servem de base para a análise de desempenho e viabilidade técnica apresentada nos capítulos seguintes.

A seguir, são detalhados os procedimentos realizados e os resultados obtidos para cada teste pertencente a esta fase.

5.2.1 Testes de Resistência a Ataques DdoS

O teste foi realizado sobre o firewall do Raspberry Pi 5 com a mesma configuração usada nos demais cenários: política *default deny*, inspeção *stateful*,

NAT, segmentação entre LAN/DMZ/Guest e Suricata operando como IPS inline via NFQUEUE para tráfego TCP/UDP. Com isso, todo o tráfego passou pelos mecanismos de filtragem e inspeção durante os ataques. O registro de eventos por segmento e a coleta de CPU, RAM e temperatura também permaneceram ativos para permitir correlação com o comportamento sob carga.

O objetivo foi medir a capacidade do firewall de permanecer disponível e retornar ao estado normal depois de ataques DDoS em camadas 3/4 e 7. As métricas acompanharam conexões por segundo suportadas, taxa de pacotes, uso de recursos e tempo de recuperação após cada ataque, alinhadas ao planejamento da Fase 2.

As ferramentas utilizadas incluíram: hping3 (SYN/UDP/ICMP flood), slowloris (DoS HTTP), nmap (varreduras agressivas), ab/wrk (carga HTTP legítima), tcpdump/wireshark para capturas, além de htop/top, Grafana e ntopng para monitoramento.

Cenário 1 - SYN Flood (L4/TCP):

- Procedimento: a partir de um host na WAN, enviou-se SYN flood para a porta HTTP de um servidor na DMZ (e, em variação, diretamente ao IP WAN do *firewall*). Em paralelo, uma carga HTTP legítima (via ab/wrk) avaliou se o serviço seguia respondendo.
- Medições registradas: contagem de conexões em conntrack, taxa de aceitação de requisições HTTP legítimas (2xx), CPU/RAM/temperatura, pacotes SYN observados (PCAP) e eventuais alertas/drops do Suricata.

Cenário 2 - UDP Flood (L3/4):

- Procedimento: geração de UDP flood com variações de tamanho de pacote (64–1024 bytes) contra a DMZ e, em variação, contra o IP WAN do *firewall*. Em paralelo, *ping* para 8.8.8.8 e curl https://www.google.com validaram o impacto em tráfego de saída legítimo.
- Medições registradas: pps recebidos (ntopng), perda nos *pings* de controle, CPU/RAM, eventos de descarte pelo *firewall*/IPS e tempo de recuperação.

Cenário 3 - ICMP Flood (Ping Flood):

- Procedimento: ICMP flood direcionado ao IP WAN do *firewall* para avaliar impacto no próprio plano de controle e efeito sobre encaminhamento de tráfego legítimo.

- Medições registradas: volume ICMP (PCAP), conntrack, CPU e disponibilidade de HTTP/HTTPS durante o ataque, além dos logs de *drop* do *firewall*.

Cenário 4 - Slowloris (L7/HTTP):

- Procedimento: ataque Slowloris contra o servidor web na DMZ, mantendo múltiplas conexões parcialmente abertas. Em paralelo, requisições HTTP bem formadas aferiram degradação do serviço sob o ataque de aplicação.
- Medições registradas: taxa de respostas 2xx/5xx do servidor, latência média das requisições legítimas, CPU/RAM no *firewall* e alertas no Suricata referentes a comportamento anômalo.

Cenário 5 - Ataque Misto/Sustentado:

- Procedimento: combinação SYN + UDP flood + Slowloris por 5–10 minutos a fim de provocar exaustão progressiva dos recursos do sistema e observar o comportamento do firewall diante de múltiplas formas de saturação.
- Medições registradas: disponibilidade de serviços (HTTP de prova), temperatura e CPU do Pi, crescimento de conntrack e eventos em *logs* (IPS/*firewall*).

Em conformidade com o plano e com a matriz de métricas do trabalho, o teste consolidou, por cenário, os seguintes indicadores:

- Conexões por segundo suportadas (CPS) e pps/Mbps efetivos do ataque observados.
- Disponibilidade do serviço-alvo (manutenção total, parcial ou indisponibilidade).
- Uso de CPU (%), uso de RAM (MB) e temperatura (°C) do Raspberry Pi durante o ataque.
- Tempo de recuperação (s) após a interrupção do ataque.
- Eventos de segurança: alertas/drops do IPS e contadores de *drop* do *firewall* associados a cada vetor.

5.2.2 Testes de Filtragem de Tráfego Malicioso

O segundo conjunto de experimentos da Fase 2 teve como objetivo validar a eficácia do sistema de detecção e prevenção de intrusões (IPS/IDS) implementado com o Suricata operando em modo inline, integrado ao firewall do Raspberry Pi via NFQUEUE. O teste buscou mensurar a capacidade do sistema em detectar e bloquear tráfego malicioso, identificando padrões típicos de varreduras, tentativas de exploração web e comunicações suspeitas, sem a utilização de *malware* real, em conformidade com as boas práticas de segurança de laboratório.

O ambiente utilizado manteve a mesma configuração de rede e política de segurança empregada nos testes anteriores, com segmentação entre LAN, DMZ e Guest, regras *default deny*, NAT para saída e encaminhamento do tráfego TCP/UDP para o IPS. O conjunto de regras ET Open foi atualizado e ajustado para operar com ações de *alert* e *drop* conforme o tipo de evento, garantindo o equilíbrio entre eficácia e minimização de falsos positivos.

Durante os testes, foram utilizados os seguintes instrumentos e ferramentas:

- Suricata (em modo IPS inline, usado como prevenção à intrusão) com logs em formato EVE JSON;
- nftables com integração via queue num 0;
- nmap, curl e hping3 para geração de tráfego de ataque e simulação de varreduras;
- jq e tcpdump para análise de logs e pacotes;
- htop, Grafana e ntopng para monitoramento de recursos e tráfego.

A telemetria do Raspberry Pi (CPU, RAM e temperatura) foi registrada em paralelo à coleta dos alertas do Suricata, permitindo correlacionar o impacto dos ataques simulados com o consumo de recursos do dispositivo.

Cenário 1 - Varredura de Portas (Port Scanning)**:**

Neste cenário, avaliou-se a capacidade do Suricata em detectar escaneamentos de portas TCP e UDP originados da WAN contra o servidor web na DMZ e contra o próprio firewall. Foram realizadas varreduras utilizando técnicas comuns (SYN scan, UDP scan, Xmas scan e version detection) com a ferramenta nmap.

Durante a execução, o Suricata registrou assinaturas da categoria “ET SCAN”, gerando alertas quase imediatos após o início das tentativas. Em modo

drop, observou-se o bloqueio parcial dos pacotes de varredura, enquanto o serviço legítimo na DMZ (HTTP) permaneceu disponível. Os registros de log incluíram o tipo de assinatura, o endereço IP de origem e a ação tomada (alerta ou bloqueio).

Foram anotadas também as variações de CPU e memória, a contagem de conexões no *conntrack* antes e após o ataque, e o tempo de resposta do sistema.

Cenário 2 - Tentativas de Exploração Web (SQL Injection e XSS):

Neste cenário, o objetivo foi verificar se o IPS seria capaz de identificar padrões clássicos de injeção SQL e Cross-Site Scripting (XSS), enviados a partir da rede LAN para o servidor web localizado na DMZ. Foram utilizados comandos curl contendo parâmetros manipulados, como ' OR 1=1-- e <script>alert(1)</script>, além de requisições com *User-Agent* típicos de ferramentas ofensivas (ex.: sqlmap/1.7.0).

Cenário 3 - Detecção de Arquivo EICAR (Malware de Teste):

Para verificar a inspeção de conteúdo, foi realizado o download do arquivo de teste EICAR (inofensivo e amplamente usado para validação antivírus) a partir de uma estação na LAN. Já em acesso via HTTPS, sem decriptação (SSL bump desativado), o conteúdo não pôde ser analisado, registrando-se a limitação do IPS quanto à inspeção de tráfego criptografado, observação que foi documentada para análise posterior.

Cenário 4 - Bloqueio de Domínios e URLs de Risco:

Este cenário teve como objetivo validar a integração entre o IPS e o proxy de filtragem web (Squid/SquidGuard) para bloquear acessos a domínios classificados como maliciosos ou de phishing, foram inseridos domínios de teste em listas locais de bloqueio.

Cenário 5 - Padrões de Comunicação Suspeita (C2/Exfiltração):

Nesta etapa, foram simuladas comunicações de comando e controle (C2) e tentativas de exfiltração de dados, utilizando URIs e *headers* com padrões comuns (por exemplo, cmd=whoami, beacon/checkin, e X-Forwarded-For: 127.0.0.1).

Cenário 6 - Bloqueio de Protocolos Inseguros:

Por fim, verificou-se o bloqueio de portas e protocolos considerados inseguros (Telnet, SMB, FTP) entre as redes Guest, LAN e DMZ. As conexões de teste foram negadas de acordo com as regras de firewall e geraram eventos de *drop* nos logs do nftables, confirmando a coerência da política de segurança segmentada.

Durante os experimentos, foram coletadas as seguintes métricas principais:

- Taxa de detecção: proporção de eventos identificados em relação às tentativas geradas;
- Taxa de bloqueio: percentual de pacotes efetivamente descartados pelo IPS/firewall;
- Falsos positivos: número de alertas em tráfego legítimo;
- Tempo de resposta: intervalo entre o início da atividade e o primeiro alerta;
- Uso médio de CPU e RAM, além da temperatura máxima do Raspberry Pi;
- Disponibilidade dos serviços legítimos (HTTP/ICMP) durante os testes.

5.2.3 Testes de Filtragem Web

Este conjunto de experimentos teve como finalidade verificar, em ambiente de laboratório, o funcionamento da filtragem de conteúdo web implementada no firewall do Raspberry Pi, cobrindo bloqueio por categorias (ex.: adulto, apostas, malware), regras por horário, bloqueio de tipos de arquivo (executáveis) e mecanismos anti-bypass. O teste foi planejado na Fase 2 (Segurança e Resiliência) do plano de testes e faz parte das funcionalidades previstas para a solução (proxy com Squid/SquidGuard/ufdbGuard).

A filtragem foi realizada com Squid operando em dois modos: proxy explícito (porta 3128) e transparente (porta 3129), este último alimentado por redirecionamento de tráfego HTTP aplicado no firewall via nftables. O ambiente manteve a segmentação de rede e política default deny já utilizadas nos demais testes, com NAT e integração das cadeias de filtro; o redirecionamento HTTP para porta 3129, foi confirmado na tabela proxy_redir. As listas e políticas do SquidGuard/ufdbGuard foram carregadas para categorização por domínios/URLs.

Ferramentas empregadas: curl/wget e navegador (para geração de requisições HTTP/HTTPS), tcpdump (amostras), htop/top e sensores do sistema (CPU/RAM/temperatura), além de coleta de logs do Squid e do mecanismo de categorização. O escopo e as métricas seguiram o plano: taxa de bloqueio correto por categoria, falsos positivos/negativos, latência adicional e benefício de cache.

Foram realizados dez cenários, cada um com roteiro padronizado (pré-checagens, execução, coleta de logs e telemetria):

1. Baseline e encanamento (controle). Sem políticas de bloqueio, mediu-se a latência de acesso via HTTP transparente e via proxy explícito (HTTP e

HTTPS). Em sequência, repetiu-se uma mesma URL para observar HIT de cache (cabeçalho X-Cache/marcações HIT/MISS no access.log).

2. Bloqueio por categoria (Adulto, Apostas, Malware). Após atualização das blocklists, um conjunto de URLs categorizadas foi requisitado em HTTP (transparente) e HTTPS (via proxy explícito). Os resultados (permitido/bloqueado), códigos HTTP e páginas de bloqueio foram registrados, juntamente com o motivo/categoria nos logs.
3. Malware seguro (EICAR) e phishing de teste. Foi tentado o download do arquivo-teste EICAR por HTTP e HTTPS (via proxy explícito). Em HTTP, esperou-se bloqueio por categoria “malware”; em HTTPS, o comportamento ficou condicionado à política adotada (SNI/SSL Bump).
4. Bloqueio de executáveis. Objetos .exe/.msi hospedados em servidor de teste foram solicitados (HTTP/HTTPS). Verificou-se o bloqueio por MIME/regex e o registro do motivo (ACL).
5. Controle por horário. Regras temporais foram aplicadas para restringir redes sociais e streaming durante uma janela definida. Testes dentro e fora da janela validaram bloqueio/permissão conforme política e o devido registro em log.
6. Whitelist (exceções de negócio). Domínios críticos foram inseridos em lista branca e acessados mesmo sob categoria global bloqueada e/ou em horário restrito, para verificar o predomínio da exceção e a marcação “ALLOW/whitelist” em log.
7. HTTPS/SSL Bump (quando habilitado). Com o certificado da CA instalado no cliente, acessos HTTPS a categorias bloqueadas foram repetidos via proxy explícito para comprovar bloqueio por conteúdo/URL e não apenas por SNI, além de observar HIT de cache em ativos estáticos. (Quando SSL Bump não esteve habilitado, o cenário foi registrado como N/A.)
8. Anti-bypass (DoH/QUIC/IP direto/proxy externo/VPN). Tentativas de contorno do filtro foram exercitadas: acesso por IP direto, uso de DoH (DNS over HTTPS), QUIC (UDP/443), proxies públicos e inicialização de VPNs de terceiros. Registrou-se a eficácia por vetor (bloqueio/permitido) e as lacunas para ações corretivas.
9. Fail-close/fail-open do proxy. Com o serviço do Squid parado, mediu-se o comportamento de clientes sem proxy configurado (HTTP transparente) e

com proxy explícito. O esperado foi bloqueio/indisponibilidade no transparente (fail-close) e erro de conexão no explícito.

10. Impacto de performance & cache. Conjunto de ativos estáticos foi requisitado repetidamente via proxy explícito, comparando tempo de resposta entre a primeira requisição (MISS) e subsequentes (HIT), com telemetria de CPU/RAM/temperatura do Pi durante a carga curta.

5.2.4 Testes de VPN

O último experimento da Fase 2 teve como objetivo validar o funcionamento completo do serviço de VPN implementado no Raspberry Pi 5, utilizando o WireGuard como mecanismo de tunelamento seguro para acesso remoto. O foco foi confirmar a segurança, estabilidade e desempenho do túnel, bem como o cumprimento das políticas de segmentação e roteamento definidas para os diferentes domínios de rede (LAN, DMZ e Guest).

O ambiente utilizado manteve as mesmas configurações estruturais aplicadas aos testes anteriores, com o firewall em operação sob política *default deny*, inspeção *stateful*, NAT para saída e segmentação das zonas internas. A interface wg0 foi criada no Raspberry Pi com a sub-rede 10.10.10.0/24, e a porta UDP 51820 permaneceu aberta na interface WAN para permitir conexões externas. Foram configurados clientes WireGuard em notebooks e dispositivos móveis (Android/iOS), cada um com pares de chaves individuais e perfis específicos de conexão. O nftables foi ajustado para permitir tráfego entre a VPN e as redes LAN (192.168.50.0/24) e DMZ (192.168.20.0/24), bloqueando o acesso à rede Guest (192.168.30.0/24), em conformidade com a política de isolamento da empresa fictícia.

Cenário 1 - Estabelecimento do túnel e *handshake:*

O primeiro teste verificou o estabelecimento da conexão VPN entre o cliente remoto e o servidor WireGuard no Raspberry Pi.

O serviço foi iniciado com systemctl enable --now wg-quick@wg0, e as informações do túnel foram inspecionadas por meio dos comandos wg show e journalctl -u wg-quick@wg0.

Os clientes importaram o arquivo de configuração contendo o endpoint público do Pi e os parâmetros de *AllowedIPs*, e em seguida realizaram o *handshake*. O monitoramento via ss -nputw confirmou a escuta ativa na porta UDP/51820.

Cenário 2 - Roteamento e segmentação (VPN → LAN/DMZ, bloqueio a Guest):

Neste cenário foi verificado se o tráfego da VPN obedecia à segmentação interna do firewall.

A partir do cliente conectado via túnel, foram realizados pings e requisições HTTP para hosts nas redes LAN e DMZ (ex.: 192.168.50.10 e 192.168.20.10), que responderam corretamente.

Em seguida, tentou-se o acesso a um host na rede Guest (192.168.30.10), o qual foi bloqueado conforme esperado, gerando logs de DROP_VPN e DROP_FORWARD.

Os resultados confirmaram o roteamento controlado da VPN e a coerência das regras de segurança aplicadas.

Cenário 3 - DNS interno e política de *split/full tunnel:*

O terceiro experimento testou o comportamento do DNS e o tipo de roteamento adotado conforme o modo de túnel.

No modo split tunnel, apenas os endereços internos (LAN e DMZ) eram encaminhados pela VPN, mantendo o tráfego comum diretamente via rede local do cliente. Nesse caso, o IP público observado permaneceu o do cliente, e as consultas DNS internas funcionaram corretamente.

No modo full tunnel, todo o tráfego passou pela VPN; o endereço IP público passou a ser o da WAN do Raspberry Pi, confirmando o roteamento completo e a tradução NAT no túnel.

Foram verificados também possíveis *DNS leaks*, sem ocorrência de vazamentos de consultas externas.

Cenário 4 - Desempenho da VPN (throughput e latência):

Para medir o desempenho do túnel, foram realizados testes com o iperf3 entre um servidor na LAN e um cliente remoto conectado via VPN. Os testes TCP e UDP, unidirecionais e bidirecionais, executados por 60 segundos cada, permitiram comparar o throughput via VPN com o baseline sem túnel.

Cenário 5 - Múltiplos peers e concorrência:

Neste cenário foram conectados três clientes VPN simultâneos (notebook, smartphone e outro PC).

Os três executaram iperf3 concorrente contra o mesmo servidor interno, enquanto o Pi manteve monitoramento contínuo de CPU e *handshakes*. O wg show indicou handshakes atualizados para todos os peers, validando a escalabilidade prática do serviço.

Cenário 6 - Mobilidade e reconexão:

Foi testada a capacidade de reconexão automática (roaming) quando o cliente mudava de rede (por exemplo, de Wi-Fi para 4G). Durante um *ping* contínuo para o gateway LAN (192.168.50.1), observou-se uma breve interrupção de 2 a 5 segundos, após a qual o túnel se restabeleceu automaticamente sem intervenção manual.

O *handshake* foi atualizado conforme previsto, demonstrando a resiliência do protocolo WireGuard em cenários de mobilidade.

Cenário 7 - Revogação de acesso:

O teste seguinte avaliou o procedimento de revogação imediata de um peer. A remoção da chave pública de um cliente em tempo real (wg set wg0 peer <PUBKEY> remove) fez com que o acesso fosse interrompido instantaneamente. Os logs confirmaram eventos de *drop* para o endereço do cliente revogado, e o *peer* não conseguiu reconectar sem reinstalação da chave.

Cenário 8 - Robustez a ruído na porta da VPN:

Por fim, avaliou-se a robustez do serviço sob tráfego espúrio na porta UDP/51820.

Um *flood* UDP foi gerado de uma máquina da WAN enquanto um cliente legítimo permanecia conectado.

Durante todos os cenários, foram coletadas métricas quantitativas em planilhas de acompanhamento, contemplando:

- Tempo de estabelecimento e reconexão;
- Disponibilidade e acesso permitido/bloqueado por rede (LAN, DMZ, Guest);
- Tipo de túnel (split/full) e endereço público observado;
- Throughput TCP/UDP, perda (%) e jitter (ms);
- Número de peers simultâneos;
- Uso de CPU e temperatura máxima do Raspberry Pi;

- Eventos de log (handshakes, drops, bytes transmitidos e recebidos).

5.3 Testes de Consumo de Recursos

A terceira fase dos experimentos teve como objetivo avaliar o consumo de recursos do sistema durante a operação do firewall baseado no Raspberry Pi 5, analisando o comportamento do dispositivo sob diferentes níveis de carga de trabalho. Essa etapa é necessária para determinar a eficiência operacional e a sustentabilidade do desempenho do Raspberry Pi quando utilizado como firewall corporativo de baixo custo.

Enquanto as fases anteriores concentraram-se em aspectos de desempenho de rede e resiliência frente a ataques, esta fase buscou compreender o impacto interno dessas operações sobre o hardware do dispositivo, medindo o uso do processador (CPU), da memória RAM e a temperatura do sistema. Tais parâmetros são fundamentais para avaliar a viabilidade prática da solução, uma vez que o Raspberry Pi possui recursos limitados em comparação com equipamentos corporativos dedicados.

Os testes foram conduzidos em ambiente idêntico ao das fases anteriores, com todas as funcionalidades do firewall ativas, incluindo nftables, Suricata (IPS inline), Squid/SquidGuard, WireGuard (VPN), Loki/Promtail/Grafana para telemetria e ntopng para visualização do tráfego. Essa configuração garantiu que as medições representassem o comportamento real do sistema em operação integrada.

Foram definidos três grupos principais de medições:

1. Monitoramento de CPU (Teste 3.1): Avaliou a variação do uso de processamento sob diferentes cargas de tráfego (idle, tráfego normal e pico), verificando se o Raspberry Pi mantém desempenho estável sem atingir limiares críticos que possam comprometer a inspeção de pacotes.
2. Monitoramento de Memória (Teste 3.2): Mediu o consumo de RAM durante períodos de operação contínua e com crescimento gradual de conexões simultâneas, observando se há tendência de esgotamento de memória ou *swapping* em cargas prolongadas.
3. Monitoramento de Temperatura (Teste 3.3): Registrou a temperatura do processador sob diferentes níveis de uso e duração, com o intuito de

identificar a ocorrência de *thermal throttling* e avaliar a necessidade de soluções de resfriamento adicionais.

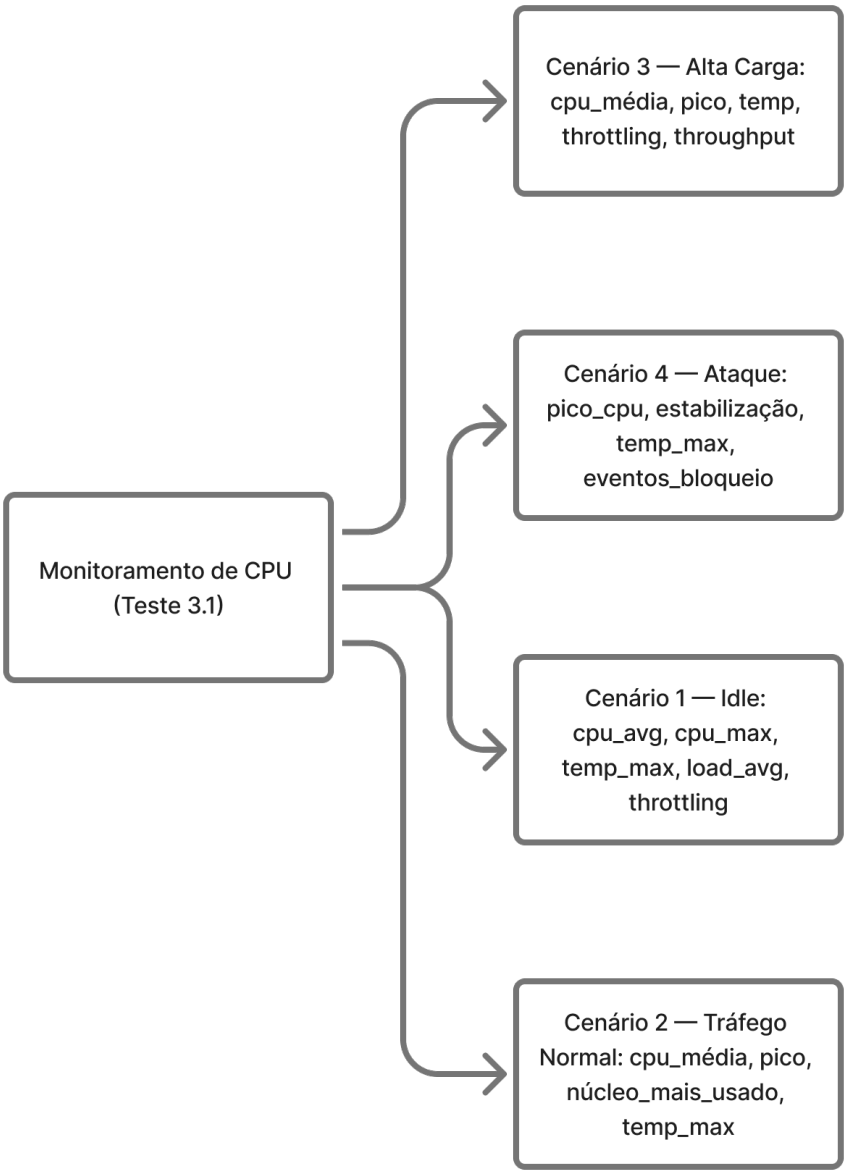
Durante todos os testes, as métricas foram coletadas por meio de ferramentas de linha de comando do sistema operacional (htop, free, vcgencmd measure_temp) e, quando aplicável, exportadas para Grafana via Promtail/Loki, permitindo análise temporal e correlação entre carga de tráfego e consumo de recursos.

Essa fase fornece subsídios para determinar se o Raspberry Pi 5 é capaz de sustentar as operações típicas de um firewall corporativo sem degradação significativa de desempenho, confirmando sua eficiência energética e operacional como alternativa de baixo custo para pequenas e médias empresas.

5.3.1 Monitoramento de CPU

O primeiro conjunto de experimentos da Fase 3 teve como finalidade avaliar o consumo de processamento (CPU) do Raspberry Pi 5 durante a operação do firewall em diferentes condições de carga, conforme mostrado na Figura 26. O objetivo foi mensurar a variação do uso do processador ao longo de cenários representativos, operação ociosa (idle), tráfego normal e situações de pico, para verificar se o dispositivo mantém desempenho estável sem atingir níveis críticos de utilização. O monitoramento foi realizado diretamente no Raspberry Pi por meio das ferramentas htop e top, complementadas por scripts de coleta periódica e, quando disponível, integração com Grafana para registro gráfico. A métrica principal observada foi o percentual de uso da CPU (%), considerando tanto o total do sistema quanto a distribuição entre os núcleos. A infraestrutura de teste manteve a mesma configuração de rede e políticas descritas anteriormente, com nftables aplicando regras de inspeção stateful, NAT, integração com o Suricata em modo IPS via NFQUEUE, e redirecionamento de tráfego HTTP/HTTPS para oproxy Squid. Esse arranjo permitiu que o monitoramento refletisse o comportamento real do firewall operando com todos os seus componentes ativos.

Figura 26 - Cenários e Métricas



Fonte: Autoria Própria

Foram definidos três cenários principais de medição:

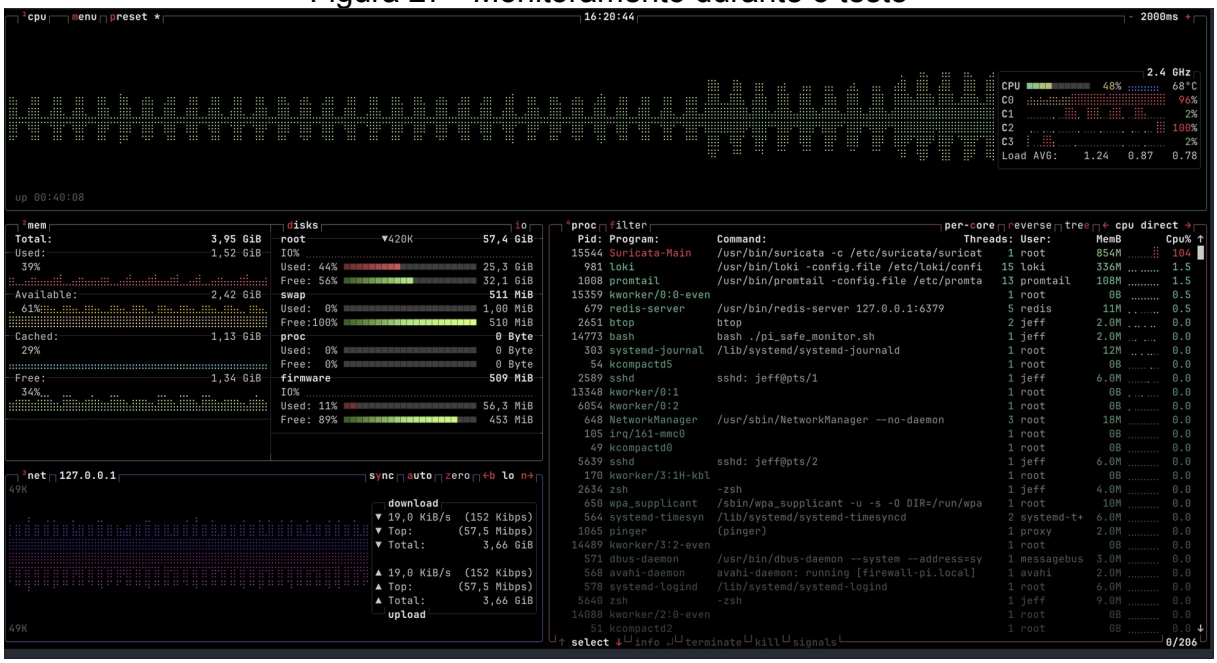
1. Operação Ociosa (Idle): O firewall permaneceu ativo, mas sem tráfego significativo atravessando as interfaces. A coleta foi feita durante 10 minutos, com

amostras registradas a cada 5 segundos. O objetivo foi estabelecer o consumo mínimo do sistema em repouso, representando o estado de espera de uma rede corporativa fora do horário de expediente.

2. Tráfego Normal Simulado: Nesse cenário, o tráfego de rede foi gerado por ferramentas como iperf3 e curl, simulando navegação web, transferências de arquivos e conexões simultâneas moderadas entre LAN e WAN. O monitoramento ocorreu por 15 minutos, enquanto os serviços de proxy e IPS processavam requisições HTTP e HTTPS legítimas. O foco foi observar o comportamento médio da CPU durante operação típica, avaliando se o uso se mantém dentro de uma faixa estável e segura (até 60–70%).

3. Cenário de Pico de Tráfego: Para avaliar o limite de processamento, reproduziram-se condições de estresse semelhantes às registradas nos testes de desempenho e segurança anteriores, geração de tráfego massivo (TCP/UDP) via iperf3 e simulações de ataque (hping3). O monitoramento foi mantido durante 5 a 10 minutos, com amostragem em intervalos curtos (5 segundos). Essa etapa permitiu identificar o pico máximo de utilização da CPU e verificar se houve ocorrência de thermal throttling (redução automática de frequência do processador por temperatura).

Figura 27 - Monitoramento durante o teste



Fonte: Autoria Própria

Durante todos os cenários, foram coletadas as seguintes métricas principais: uso médio e máximo de CPU (%), tempo total de observação (s), número de núcleos ativos, e temperatura do processador (°C), conforme observadas na Figura 27.

5.3.2 Monitoramento de Memória

O segundo conjunto de experimentos da Fase 3 teve como objetivo analisar o consumo de memória RAM do Raspberry Pi 5 durante a operação do firewall, considerando diferentes condições de carga e tempo de execução. O foco deste teste foi identificar se o dispositivo mantém estabilidade na alocação de memória e se há ocorrência de saturação, *swapping* ou degradação de desempenho sob uso prolongado.

A importância dessa avaliação reside no fato de que o Raspberry Pi, embora equipado com memória relativamente limitada em relação a equipamentos corporativos dedicados, precisa sustentar diversos serviços simultaneamente, incluindo nftables, Suricata (IPS/IDS), Squid/SquidGuard, WireGuard (VPN) e Loki/Promtail/Grafana. O comportamento eficiente na gestão de memória é, portanto, um fator determinante para a viabilidade da solução proposta.

O monitoramento foi realizado por meio das ferramentas free, vmstat, e dos dashboards do Grafana, que receberam dados do Promtail e do Loki configurados para coleta contínua de métricas do sistema. Foram observadas as variáveis de memória total, memória usada, buffers/caches, memória livre e swap, permitindo distinguir o consumo efetivo dos processos de firewall e serviços auxiliares do uso geral do sistema.

Três cenários principais foram definidos para a coleta:

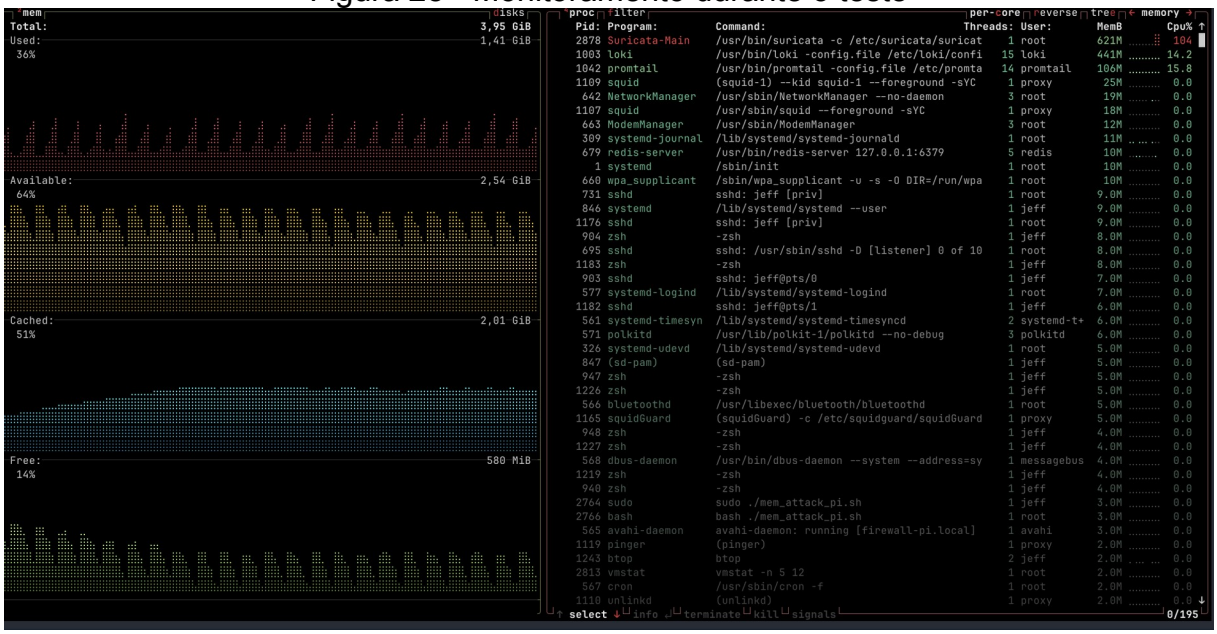
1. Operação de Base (Idle): O sistema foi mantido em funcionamento sem tráfego significativo, apenas com os serviços do firewall ativos. As medições foram realizadas a cada 10 segundos durante 10 minutos, registrando o consumo estável mínimo de RAM e a presença de caches do sistema operacional.
2. Crescimento Gradual de Conexões: Neste cenário, foram simuladas conexões TCP progressivas utilizando Apache Bench (ab) e wrk, conforme metodologia dos testes de desempenho anteriores. O objetivo foi observar o comportamento da utilização de memória à medida que aumentava o

número de sessões simultâneas (100, 500, 1000, 5000, até 10000 conexões). O conntrack-tools foi utilizado para correlacionar a variação do número de conexões com o crescimento no consumo de RAM. As leituras foram coletadas continuamente por 20 minutos.

3. Operação Prolongada (Estresse Térmico e Sustentação): Com tráfego misto (HTTP, UDP e VPN) mantido por um período de 6 a 8 horas, avaliou-se se o consumo de memória apresentava crescimento constante (*memory leak*) ou se permanecia estável ao longo do tempo. Essa etapa também permitiu verificar o comportamento do cache do sistema e da área de swap em operação contínua.

Durante todos os experimentos, foram coletadas as métricas principais: memória usada (MB), percentual de utilização (%), uso de swap, tempo de observação (s) e temperatura do processador (°C) para correlacionar eventuais aumentos de uso com o aquecimento do sistema, conforme observado na Figura 28. As informações foram armazenadas em planilhas de acompanhamento e visualizadas por gráficos no Grafana, possibilitando análise temporal detalhada.

Figura 28 - Monitoramento durante o teste



Fonte: Autoria Própria

5.3.3 Monitoramento de Temperatura

O terceiro conjunto de experimentos da Fase 3 teve como propósito avaliar o comportamento térmico do processador do Raspberry Pi 5 durante a operação do firewall em diferentes condições de carga, a fim de verificar se o dispositivo sofre com elevação excessiva de temperatura ou ativação de thermal *throttling* (redução automática de frequência de CPU para evitar superaquecimento).

A análise da temperatura é um aspecto crítico neste tipo de solução, uma vez que o Raspberry Pi, apesar de compacto e eficiente energeticamente, não dispõe de um sistema ativo de refrigeração industrial como os equipamentos corporativos. Assim, compreender o comportamento térmico sob diferentes cenários de operação permite determinar se a utilização contínua como *firewall* é segura e sustentável em longo prazo.

O monitoramento foi realizado por meio do comando vcgencmd measure_temp, que fornece leituras diretas do sensor interno da CPU, complementado por coletas automatizadas via script em intervalos de 5 segundos e integração com o Grafana, que apresentou gráficos em tempo real de temperatura e frequência de clock. Também foram observadas as variações de temperatura em conjunto com o uso de CPU e memória, para correlacionar picos térmicos com momentos de maior carga de processamento.

Três cenários principais foram definidos:

1. Operação Normal (Idle e Tráfego Moderado): O firewall permaneceu operando em regime normal, processando tráfego básico da LAN e dos serviços internos (proxy, IPS e VPN). Foram coletadas leituras de temperatura durante 30 minutos, registrando a faixa térmica típica do sistema em uso cotidiano, sem picos de carga significativos.
2. Cenário de Carga Máxima: Para simular a condição mais exigente, foram reproduzidos os mesmos testes de alto volume utilizados na Fase 1 (throughput) e na Fase 2 (ataques DDoS e filtragem IPS). Durante 15 minutos de tráfego intenso, a temperatura foi medida continuamente, observando a reação do sistema de resfriamento passivo do Raspberry Pi e a possível ativação de throttling. O comando cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_cur_freq foi utilizado para monitorar a frequência de operação dos núcleos.

3. Operação Prolongada (Teste de Sustentação): Nesse cenário, o firewall foi mantido em funcionamento contínuo por mais de 8 horas com tráfego moderado e todos os serviços ativos. O objetivo foi identificar padrões de aumento gradual de temperatura (indicando acúmulo térmico) e eventuais variações correlacionadas a ciclos de uso da CPU.

Durante a execução, as métricas registradas incluíram: temperatura média (°C), temperatura máxima (°C), frequência de clock (MHz), uso médio de CPU (%), tempo de observação (s) e ocorrência de throttling. As leituras foram armazenadas em planilhas e representadas graficamente para análise visual no Grafana.

5.4 Testes de Estabilidade e Escalabilidade

A quarta fase dos experimentos teve como objetivo verificar a confiabilidade operacional e a capacidade de crescimento do firewall baseado no Raspberry Pi 5 quando submetido a operações contínuas e a aumento progressivo de demanda. Enquanto as fases anteriores mediram desempenho instantâneo, segurança e consumo de recursos, esta etapa concentrou-se em como o sistema se comporta ao longo do tempo e quais são seus limites práticos em cenários de expansão típicos de uma PME.

Foram avaliados três aspectos complementares: (i) estabilidade de uptime sob operação ininterrupta; (ii) tendência de degradação de performance após longos períodos (efeitos cumulativos de aquecimento, *memory leaks* ou saturação de buffers/conntrack); e (iii) escalabilidade diante do crescimento de dispositivos e de tráfego. A medição dessas dimensões permite aferir se a solução mantém um nível de serviço consistente sem reinicializações não planejadas, sem quedas acentuadas de throughput/latência e com margem para expansão de usuários e fluxos.

O ambiente de testes manteve toda a pilha funcional ativa, *nftables* com política *default deny* e inspeção *stateful*, Suricata (IPS inline), Squid/SquidGuard, WireGuard (VPN) e os serviços de telemetria e logs. A instrumentação consolidou as coletas usadas nas fases anteriores: Promtail/Loki/Grafana para séries temporais (CPU, RAM, temperatura, tráfego, eventos), ntopng para visibilidade de rede, conntrack-tools para a tabela de estados e contadores do nftables/IPS para quedas e bloqueios. Esse arranjo possibilitou correlacionar indicadores de recurso

(CPU/RAM/°C) com métricas de rede (throughput, latência, perda) ao longo de períodos estendidos.

A fase foi estruturada nos seguintes testes:

1. 5.4.1 Teste 4.1 - Uptime contínuo: Verifica a estabilidade operacional do firewall durante um período prolongado de funcionamento ininterrupto, monitorando indisponibilidades, reinicializações inesperadas, falhas de serviço (proxy/IPS/VPN) e eventuais eventos de falha térmica. São registrados *health checks* periódicos (ping/HTTP de controle), disponibilidade dos *daemons* e evolução de CPU/RAM/temperatura.
2. 5.4.2 Teste 4.2 - Degradação de performance ao longo do tempo: Mede variações de throughput e latência em janelas regulares (ex.: ciclos horários) sob carga controlada e repetível, comparando com os valores de referência. O objetivo é identificar tendências de queda (ex.: aquecimento acumulado ou fragmentação de recursos) e sua correlação com telemetria e contadores do sistema.
3. 5.4.3 Teste 4.3 - Escalabilidade: Avalia a capacidade de crescer em número de dispositivos/fluxos e volume de tráfego, de forma escalonada, até a proximidade do limite aceitável. São observadas entradas ativas no conntrack, uso de CPU por núcleo, impacto em latência e o ponto em que surgem erros, *timeouts* ou quedas perceptíveis.

Em todos os testes, foram consolidadas: horas de uptime sem falha, variação percentual de throughput/latência entre início e fim das janelas, picos e médias de CPU/RAM/temperatura, entradas conntrack máx., eventos de drop/alerta por período e ocorrências de falha de serviço (se houver).

Com essa estrutura, a Fase 4 estabelece se o Raspberry Pi 5 sustenta operação contínua e acompanha o crescimento de uma rede de pequeno/médio porte sem comprometer a qualidade do serviço, apontando limites práticos e condições operacionais para adoção segura.

5.4.1 Testes de Uptime

Este teste teve como objetivo avaliar a estabilidade operacional contínua do firewall implementado no Raspberry Pi 5, verificando a sua capacidade de permanecer em operação ininterrupta com todos os serviços ativos (nftables,

Suricata/IPS, Squid/SquidGuard, WireGuard/VPN e telemetria) durante um período prolongado. Pretendeu-se observar a ocorrência de indisponibilidades, reinicializações não planejadas, quedas de serviços específicos e eventuais degradações progressivas associadas a uso de CPU, memória, temperatura e crescimento da tabela de estados (conntrack). Considerou-se falha qualquer interrupção no tráfego ICMP superior a 3 segundos ou reinicialização espontânea do sistema.

Ambiente e instrumentação. O teste foi conduzido sobre a mesma topologia e políticas descritas nos capítulos anteriores, com coleta contínua por Promtail/Loki/Grafana (séries temporais de CPU, RAM, temperatura e tráfego), apoio do ntopng (visibilidade de rede) e observabilidade de serviços via journalctl/systemd. Os seguintes *health checks* automáticos foram executados em intervalos de 60 segundos por *cron* durante todo o período:

- ICMP externo (WAN): ping para um destino público (controle de conectividade geral).
- HTTP DMZ: requisição a um servidor web de teste (esperado HTTP 200).
- HTTP via proxy: requisição via Squid (verificando HIT/MISS e latência básica).
- DNS: consulta dig a um resolvedor interno/forwarder (esperado NOERROR).
- VPN: verificação de *handshake/bytes* em wg show (status up).
- Tabela de estados: amostra de conntrack -C e contadores do nftables/IPS.

Duração e carga de fundo. O dispositivo permaneceu em operação por 7 dias consecutivos (≈168 h). Para evitar um cenário puramente ocioso, foi aplicado tráfego de fundo representativo do horário comercial (navegação web, transferências moderadas e sessões VPN), mantendo janelas diárias com maior atividade e períodos de baixa carga fora do expediente.

Procedimento:

1. Sincronização de horário (NTP) e validação dos *health checks* por 30 minutos (janela de aquecimento).
2. Início do período de observação contínua (7 dias) com métricas coletadas a cada 60 s e *snapshots* de sistema (CPU/RAM/temperatura/frequência) a cada 5 min.

3. Auditoria diária de logs do kernel/nftables/IPS/proxy/VPN para identificar eventos de erro, quedas e rotações de log.
4. Consolidação automática de dados (CSV/JSON) e geração de gráficos no Grafana para correlação temporal.

Métricas registradas:

- Uptime contínuo (h) e número de incidentes (reinicializações não planejadas, quedas de serviço).
- Taxa de sucesso dos *health checks* (ICMP/HTTP/Proxy/DNS/VPN).
- CPU média e pico (%), RAM média/pico (MB), temperatura média/pico (°C).
- Entradas máximas no conntrack e contadores de drop (nftables/IPS) por dia.
- Eventos relevantes de sistema/serviço (journalctl): erros, *restarts* e avisos de recurso.

5.4.2 Testes de Degradação

Este teste teve como objetivo identificar tendências de degradação de desempenho ao longo do tempo no firewall baseado em Raspberry Pi 5, correlacionando variações de throughput, latência e perda com a telemetria de recursos (CPU, RAM e temperatura) durante períodos estendidos de operação. Diferentemente dos testes pontuais, aqui foram realizadas medições periódicas e repetíveis em janelas regulares ao longo de horas/dias, a fim de detectar efeitos cumulativos (ex.: aquecimento sustentado, crescimento da tabela de estados, fragmentação de recursos ou vazamento de memória).

Ambiente e instrumentação. O teste manteve toda a pilha funcional ativa (firewall *stateful*, IPS inline, proxy, VPN e coleta de logs/telemetria). As métricas de rede foram obtidas com iperf3 (TCP/UDP) e ping/mtr (RTT/jitter), enquanto Promtail/Loki/Grafana consolidaram séries temporais de CPU, RAM, temperatura e tráfego. conntrack-tools registrou a evolução das entradas da tabela de estados e contadores de descarte no firewall/IPS.

Metodologia geral:

1. Baseline (T0): execução do mesmo conjunto de testes de desempenho (throughput TCP/UDP, RTT ICMP) para referência inicial.

2. Coletas periódicas: repetição do mesmo roteiro a cada 1 hora nas primeiras 12 horas, depois a cada 3 horas até completar 48 horas; em janela adicional, repetiu-se por 5 dias com amostragem 2× ao dia.
3. Cargas de fundo controladas: geração de tráfego representativo em horários de “expediente” e carga reduzida fora dele, para observar o efeito do padrão de uso.
4. Correlações: para cada medição, foram capturados CPU/RAM/°C e conntrack -C, além de anotações sobre eventos de sistema (rotação de logs, atualização de listas do proxy, reinícios planejados de serviços, se houver).

Cenários executados:

1. Cenário 1: Carga moderada e contínua (perfil horário comercial)
   - *Procedimento:* tráfego TCP/UDP estável de 80–150 Mbps entre LAN↔WAN, com navegação via proxy e IPS ativo; medições de throughput e RTT repetidas por 24 h.
   - *Métricas:* variação percentual de throughput em relação ao T0, RTT médio/máx., jitter, CPU/RAM/°C e conntrack.
2. Cenário 2: Sustentação sob carga alta (janela concentrada)
   - *Procedimento:* janelas de 60–90 min com picos de 300–400 Mbps (múltiplos *streams* TCP e UDP concorrentes), repetidas ao longo de 8 h; medições a cada 30 min.
   - *Métricas:* throughput médio por janela, perda (UDP), evolução térmica, CPU pico por núcleo e eventuais drops do firewall/IPS.
3. Cenário 3: Ciclo dia/noite e eventos recorrentes de sistema
   - *Procedimento:* operação normal por 5 dias, com duas campanhas de medição/dia (manhã e fim de tarde) e marcação de eventos de manutenção rotineiros (ex.: rotação de logs, atualização de listas/certificados).
   - *Métricas:* desvio percentual de throughput e RTT entre as campanhas do dia 1 e as do dia 5; registros de estabilidade de serviços.

Métricas consolidadas:

- Degradação de throughput (%): [(valor atual – baseline) / baseline × 100].
- Variação de latência (ms e %): ΔRTT médio/máx. versus baseline.
- Perda/jitter (UDP): percentual e ms em janelas fixas.

- CPU média/pico (%), RAM média/pico (MB), temperatura média/pico (°C).
- conntrack máx. (entradas simultâneas) e contadores de drop por período.
- Eventos de serviço: quedas, *restarts* e tempos de resposta de *health checks*.

5.4.3 Testes de Escalabilidade

Este teste teve como objetivo determinar os limites práticos de escala do firewall baseado em Raspberry Pi 5, considerando o crescimento do número de dispositivos, o aumento progressivo de tráfego e a pressão sobre a tabela de estados (conntrack). Buscou-se aferir em que ponto surgem sinais de degradação (elevação de latência, quedas/erros de conexão, CPU próxima ao teto, saturação de conntrack) e qual o patamar seguro de operação para uma PME.

Ambiente e instrumentação. Toda a pilha permaneceu ativa (nftables *stateful*, Suricata em modo IPS, Squid/SquidGuard, WireGuard, telemetria e logs). Para simular clientes, foram utilizados contêineres e *namespaces* conectados às VLANs internas (LAN/Guest/DMZ). A geração de tráfego utilizou wrk/ab (HTTP), iperf3 (TCP/UDP) e scripts com curl (navegação e downloads). As medições contínuas foram registradas em Grafana (CPU/RAM/°C, throughput, latência), com conntrack-tools para acompanhar entradas ativas e contadores do nftables/IPS para quedas.

Metodologia:

1. Escala por dispositivos (clientes simulados)
    - Incrementos: 10 → 25 → 50 → 100 → 150 → 200 clientes.
    - Perfil por cliente: *mix* leve (navegação HTTP via proxy, DNS, *keep-alive* ICMP) + 1 fluxo iperf3 de baixa taxa intermitente.
    - Medições a cada 5 minutos: RTT interno (LAN↔DMZ), tempo de resposta HTTP, uso de CPU/RAM/°C e conntrack -C.
2. Escala por conexões/fluxos
    - Para 100 clientes, ramp-up de conexões HTTP persistentes: 1, 5, 10, 25, 50 e 100 conexões por cliente (wrk/ab), acompanhando latência p50/p95, erros (5xx/timeout) e crescimento do conntrack.
    - Fase adicional com iperf3 UDP para observar perda/jitter sob alta concorrência.
3. Escala de peers VPN (variação)

- Conexões WireGuard simultâneas: 1, 3, 5, 8, 10 peers.
- Cada peer executou um perfil leve (HTTP/DNS) e, em janelas, fluxos iperf3 TCP concorrentes.
- Observou-se throughput agregado, CPU por núcleo, bytes/handshakes do wg show.

Métricas coletadas:

- Clientes ativos e conexões simultâneas (totais e por cliente).
- Throughput agregado (LAN↔WAN) para TCP/UDP; perda e jitter (UDP).
- Latência HTTP (p50/p95) via proxy e RTT ICMP entre segmentos.
- Entradas conntrack (valor e % do limite) e contadores de drop.
- CPU média/pico (%), RAM média/pico (MB), temperatura média/pico (°C).
- Para VPN: nº de peers, throughput por peer e agregado, handshakes.

6 ANÁLISE DOS RESULTADOS

Este capítulo apresenta a discussão dos dados obtidos durante a fase experimental. A análise foi estruturada para responder aos objetivos específicos do trabalho, agrupando os resultados por dimensões críticas de avaliação: desempenho de rede, segurança e resiliência, consumo de recursos computacionais e, por fim, a estabilidade e escalabilidade da solução em longo prazo.

Os dados aqui interpretados buscam validar a hipótese central de que o Raspberry Pi 5, apesar de suas limitações de hardware inerentes a um computador de placa única (SBC), oferece capacidade técnica suficiente para atuar como *firewall* de borda em redes de pequeno e médio porte.

Por meio dessa análise comparativa e interpretativa, busca-se responder à questão central deste trabalho: avaliar se o Raspberry Pi pode ser considerado uma alternativa viável e eficaz a firewalls corporativos de maior custo, especialmente em ambientes de pequeno e médio porte.

6.1 Análise do Desempenho de Rede

Esta seção avalia a capacidade do dispositivo em processar e encaminhar tráfego, focando na experiência do usuário final e na eficiência do throughput.

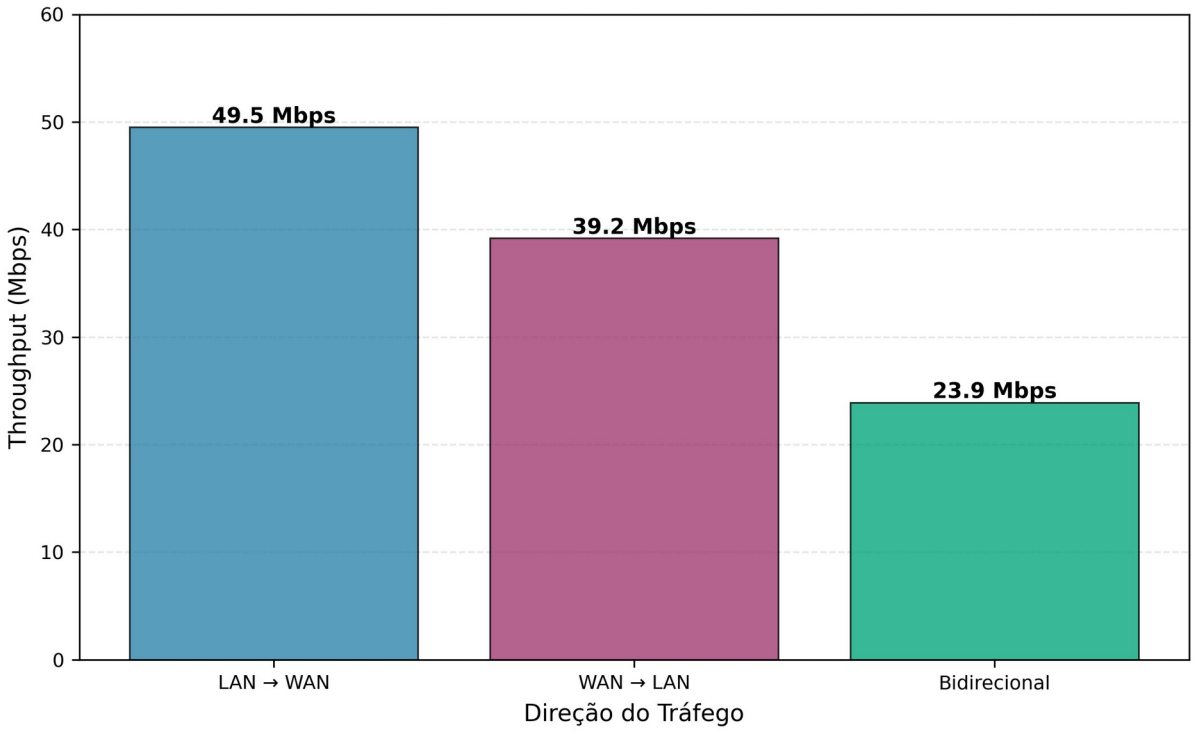
6.1.1 Throughput e Capacidade de Vazão

A avaliação do throughput, baseada nos dados dos testes, demonstrou que o Raspberry Pi 5 é capaz de sustentar taxas de transferência compatíveis com planos de internet corporativos de entrada e intermediários.

Nos testes TCP, o dispositivo atingiu taxas médias entre 23,9 Mbps e 49,5 Mbps em fluxos unidirecionais, com picos próximos a 50 Mbps. A Figura 29 apresenta os resultados de throughput TCP nas três direções testadas. Embora abaixo do limite teórico da interface Gigabit, esse desempenho reflete o custo computacional da inspeção de estado (*stateful inspection*) em software. Durante estes testes, o uso médio da CPU permaneceu controlado, variando entre 14% e 24%, e a temperatura oscilou entre 61°C e 64°C, evidenciando que o processador não foi o gargalo primário para fluxos TCP simples. A perda de pacotes manteve-se

em 1,6%, um valor aceitável e compensado pelos mecanismos nativos de retransmissão e controle de congestionamento do protocolo TCP.

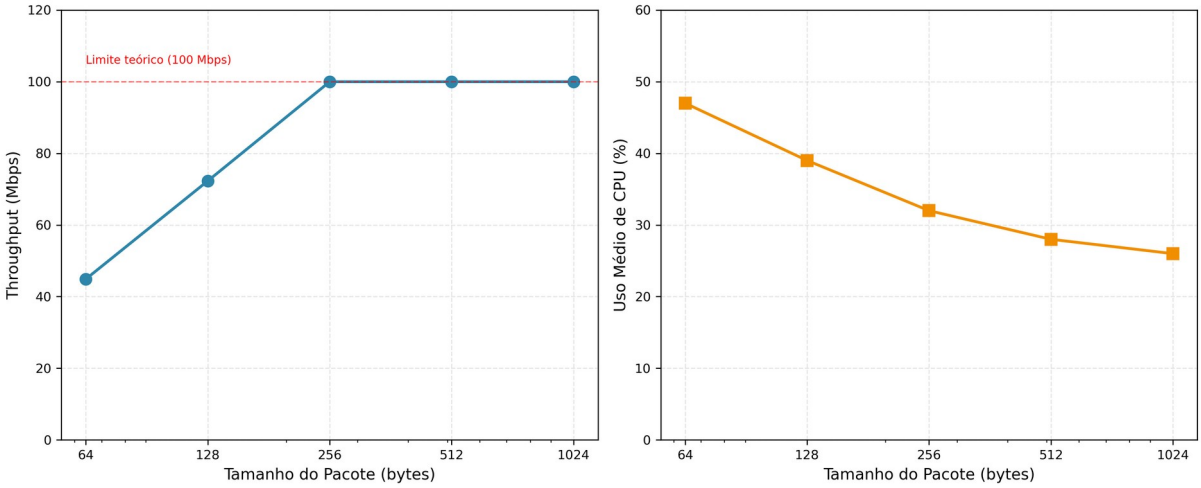
Figura 29 - Throughput médio TCP em três direções



Fonte: Autoria Própria

Nos testes UDP, realizados com variações no tamanho dos pacotes (de 64 a 1024 bytes), constatou-se um comportamento esperado: o throughput aumentou proporcionalmente ao tamanho dos pacotes, alcançando 100 Mbps a partir de 256 bytes. O throughput aumentou proporcionalmente ao tamanho dos pacotes, alcançando o máximo desempenho a partir de 256 bytes. Com pacotes menores (64 bytes), o throughput médio caiu para 44,9 Mbps, devido ao maior overhead de processamento e ao número elevado de interrupções por pacote. A Figura 30 demonstra a correlação entre tamanho de pacote UDP e eficiência do sistema. O uso médio de CPU variou entre 26 % e 47 %, atingindo o pico nos cenários com pacotes pequenos, o que reforça o impacto do tamanho do pacote na eficiência do processamento. A temperatura máxima registrada foi de 70 °C, dentro dos limites de operação seguros do dispositivo. Apesar da perda de pacotes moderada (48 a 71 pacotes por ciclo de teste), os resultados demonstram que o Raspberry Pi 5 conseguiu manter alta taxa de entrega e estabilidade térmica.

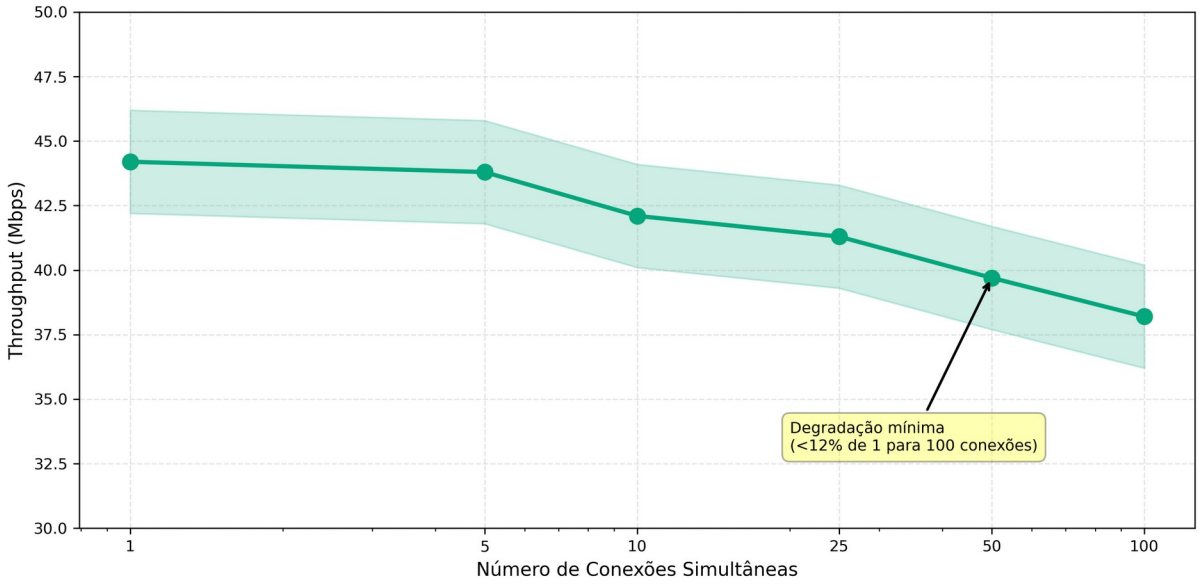
Figura 30 - Relação entre tamanho de pacote UDP, throughput e uso de CPU



Fonte: Autoria Própria

Em cenários de múltiplas conexões simultâneas, o dispositivo demonstrou estabilidade. Mesmo sob a carga de 50 fluxos paralelos, o throughput médio se manteve estável entre 39 Mbps e 44 Mbps. O monitoramento de recursos durante este teste revelou um consumo de CPU entre 19% e 22% e uso de memória RAM estável em 2,1 GB. Isso demonstra que o gerenciamento da tabela de estados (*conntrack*) foi eficiente e não introduziu latência excessiva de busca nem degradação de performance, validando a escalabilidade do dispositivo para ambientes multiusuário. A Figura 31 demonstra a estabilidade do throughput com 1 a 100 conexões simultâneas, apresentando degradação inferior a 12%, o que valida a aplicabilidade em ambientes multi-usuário.

Figura 31 - Estabilidade do throughput com conexões simultâneas crescentes



Fonte: Autoria Própria

6.1.2 Latência e Tempo de Processamento

O Teste 1.2 teve como objetivo avaliar a latência introduzida pelo firewall implementado no Raspberry Pi 5 em diferentes condições de operação, considerando desde o comportamento básico do encaminhamento de pacotes até situações com tráfego intenso, variação de protocolos e comunicação entre VLANs internas.

No cenário de latência base, foram enviados 1.000 pacotes ICMP de 64 bytes para um servidor externo, resultando em um RTT médio de 21,4 ms, com valores variando entre 19,7 ms e 27,3 ms, e jitter de apenas 1,8 ms. Não houve perda de pacotes, e o uso médio de CPU permaneceu em 12%, com temperatura entre 60 °C e 63 °C. Esses valores sugerem que o firewall adicionou aproximadamente 1,2 ms ao RTT típico, um impacto irrelevante em condições normais de uso.

Quando a análise passou a considerar a variação do tamanho dos pacotes, observou-se que a latência cresceu proporcionalmente ao volume de dados processados. O RTT médio saiu de 21,8 ms para pacotes de 64 bytes e atingiu 30,8 ms para pacotes de 1472 bytes, com jitter aumentando de 1,9 ms para 4,9 ms nesse mesmo intervalo. A perda de pacotes permaneceu baixa, apenas 1% nos pacotes maiores, e o uso de CPU variou entre 18% e 34%, com temperatura máxima de 67

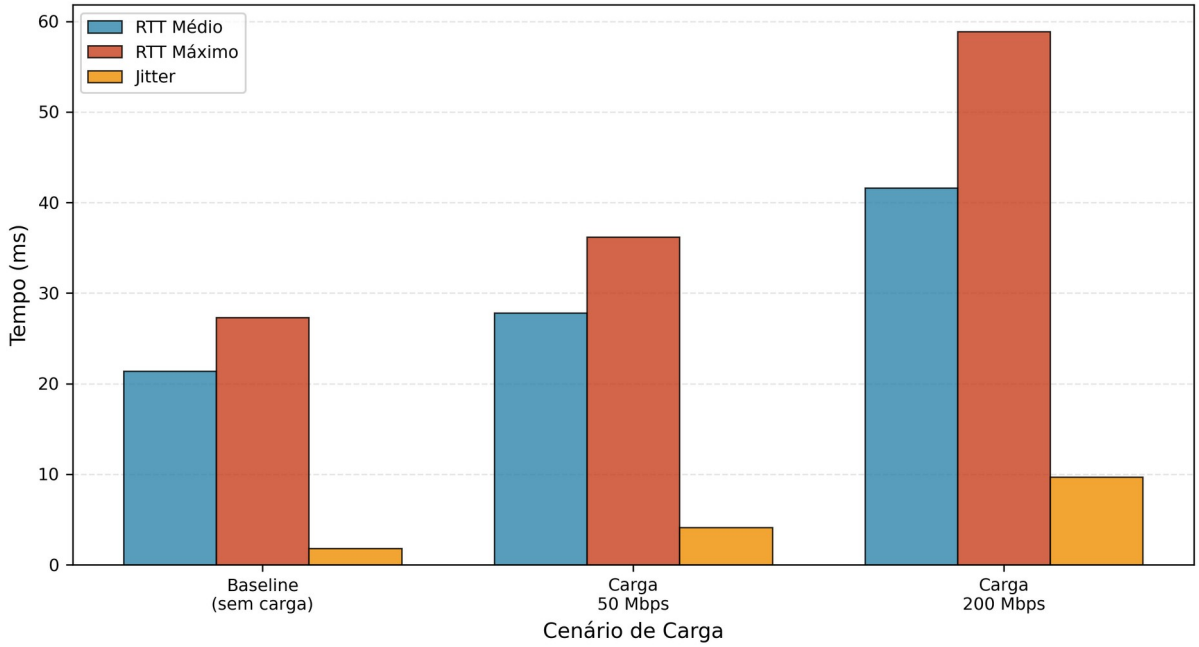
°C. Essa elevação gradual era esperada, especialmente devido à inspeção de profundidade do Suricata, que aumenta o custo computacional conforme o tamanho do pacote cresce

Na comparação entre protocolos, o comportamento se manteve dentro do previsto. O ICMP registrou RTT médio de 21,0 ms, com jitter de 1,7 ms, enquanto o TCP apresentou maior latência, chegando a 24,6 ms em média e jitter de 2,9 ms. Essa diferença decorre da necessidade de criação de estado no conntrack e da inspeção mais detalhada aplicada ao tráfego TCP. Já o UDP obteve o menor RTT médio, de 19,8 ms, ainda que com jitter levemente superior ao ICMP devido ao descarte pontual em filas internas. Em todos os casos, o uso de CPU se manteve entre 15% e 28%, sem qualquer indício de sobrecarga.

Os testes com tráfego simultâneo gerado pelo iperf3 revelaram como o firewall se comporta sob carga mais intensa, mostrado na Figura 32. Com 50 Mbps de tráfego paralelo, o RTT médio aumentou para 27,8 ms e o jitter atingiu 4,1 ms, enquanto o uso de CPU chegou a 56% em seus picos e a temperatura alcançou 70 °C. O impacto foi moderado e dentro de limites operacionais seguros. Já sob carga de 200 Mbps, o RTT subiu para 41,6 ms, com jitter de 9,7 ms e perda de 2% dos pacotes. Esses valores refletem o esforço computacional elevado observado no uso de CPU, que chegou a 92% em alguns momentos, com temperatura de 75 °C. Ainda assim, o sistema manteve estabilidade e não apresentou throttling.

Por fim, nos testes de comunicação entre VLANs internas, a latência manteve-se extremamente baixa. A comunicação entre LAN e DMZ apresentou RTT médio de 1,3 ms, enquanto o enlace entre servidores privados e a LAN registrou apenas 0,9 ms. O tráfego proveniente da VPN exibiu RTT médio de 3,8 ms devido ao overhead de criptografia, mas ainda assim dentro de valores perfeitamente aceitáveis. A comunicação entre LAN e Guest foi bloqueada, conforme definido nas regras de segmentação. O uso de CPU nesses testes variou entre 12% e 19%, com temperatura máxima de 64 °C, reforçando que o processamento interno do firewall não impõe atraso relevante ao encaminhamento local. A Figura 32 apresenta a evolução do RTT e jitter sob diferentes cargas, evidenciando que até 50 Mbps a latência permanece aceitável (<30 ms), tornando-se limitante apenas acima de 200 Mbps.

Figura 32 - RTT e jitter em diferentes cenários de carga



Fonte: Autoria Própria

6.1.3 Gerenciamento de Conexões Simultânea

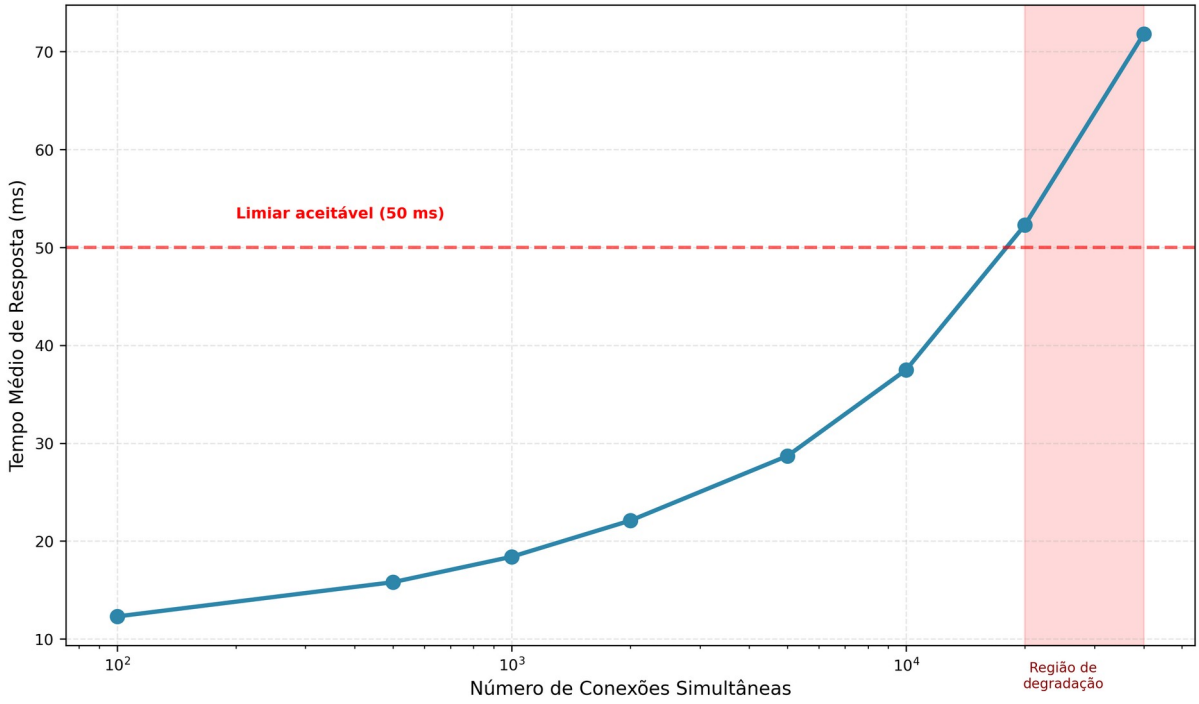
O teste de estresse da tabela de estados revelou uma capacidade de gerenciamento de sessões muito superior à demanda típica de pequenas redes. O dispositivo gerenciou com sucesso 98.980 conexões TCP simultâneas sem falhas ou resets. O dado mais relevante deste teste foi o consumo de recursos: com quase 100 mil conexões ativas, a tabela *conntrack* ocupou menos de 2% de sua capacidade total, e o uso de memória RAM manteve-se estável em 2,9 GB (Figura 35).

No entanto, o teste identificou claramente o gargalo arquitetural do sistema: enquanto a memória sobrou, a CPU atingiu 84% de utilização e a temperatura chegou a 74°C. Isso indica que o limite de escalabilidade do Raspberry Pi 5 para conexões simultâneas é definido pelo poder de processamento (CPU-bound) e não

pela memória (Memory-bound), ao contrário de muitos roteadores comerciais de entrada que sofrem com falta de RAM para grandes tabelas NAT.

A Figura 33 ilustra que o tempo de resposta cresce linearmente com o aumento do número de conexões.

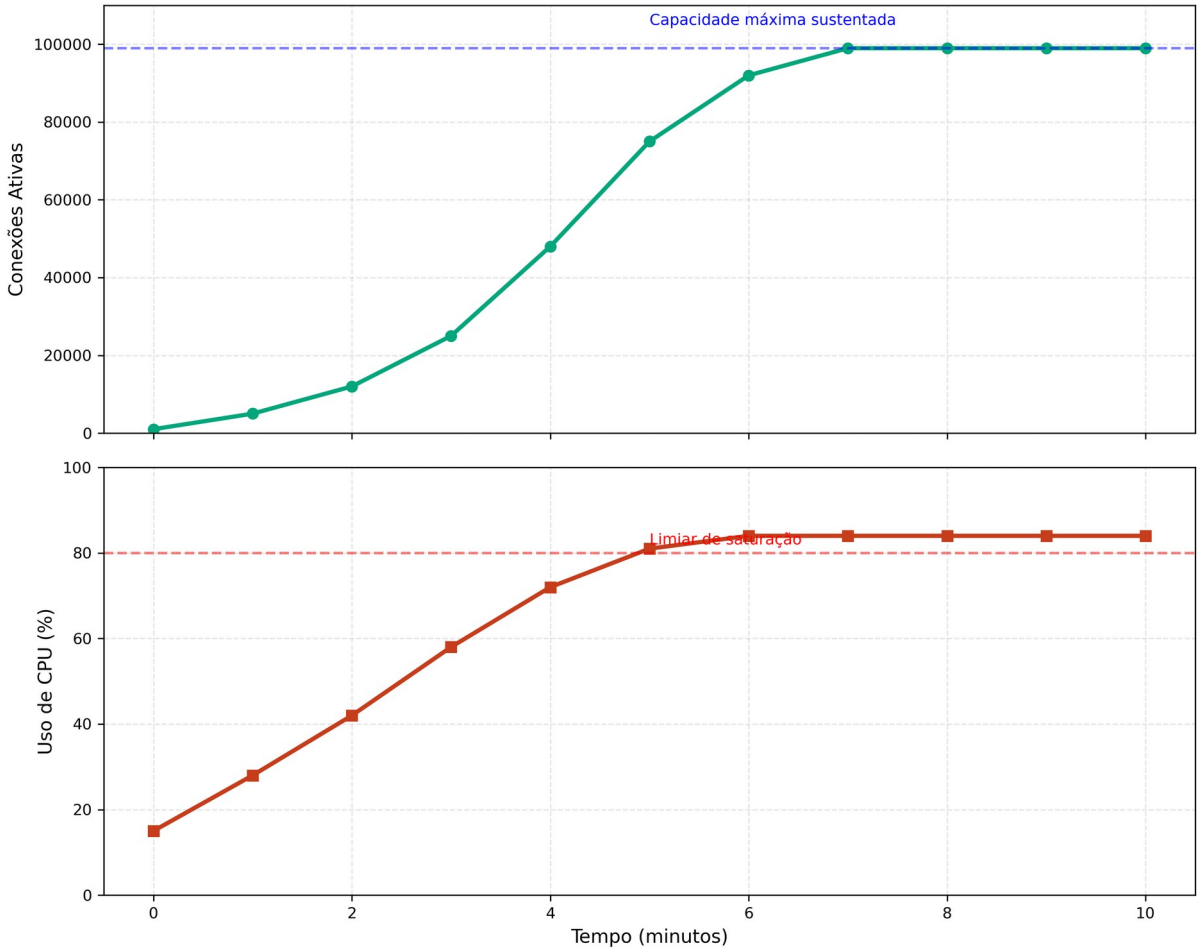
Figura 33 - Crescimento do tempo de resposta



Fonte: Autoria Própria

No teste de sustentação extrema, ao atingir 1 milhão de conexões cumulativas, a utilização da CPU chegou ao pico de 91%. Esse dado é vital para o dimensionamento da solução: o Raspberry Pi 5 é extremamente robusto para tráfego legítimo e alto volume de usuários (escalabilidade horizontal), mas sua alta dependência da CPU para gerenciar estados o torna sensível a ataques volumétricos massivos (*state exhaustion attacks*), sugerindo que, em ambientes críticos, ele deve ser complementado por proteções de borda externas. A Figura 34 mostra a sustentação de aproximadamente 100.000 conexões ativas durante 10 minutos, com CPU estável em torno de 80-90%.

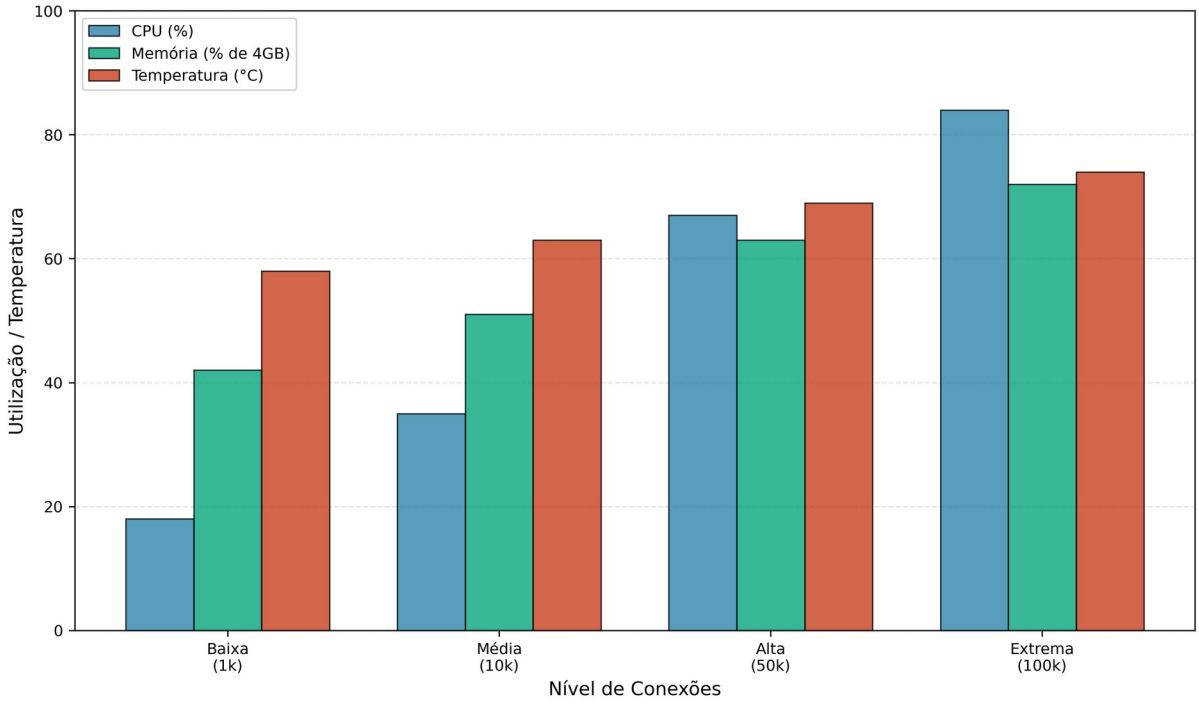
Figura 34 - Conexões ativas e CPU ao longo de 10 minutos



Fonte: Autoria Própria

A Figura 35 demonstra que a CPU é o principal limitante, atingindo 84% com 100k conexões, enquanto RAM e temperatura permanecem em níveis aceitáveis.

Figura 35 - CPU, Memória e Temperatura em diferentes níveis de conexões



Fonte: Autoria Própria

6.1.4 Confiabilidade e Perda de Pacotes

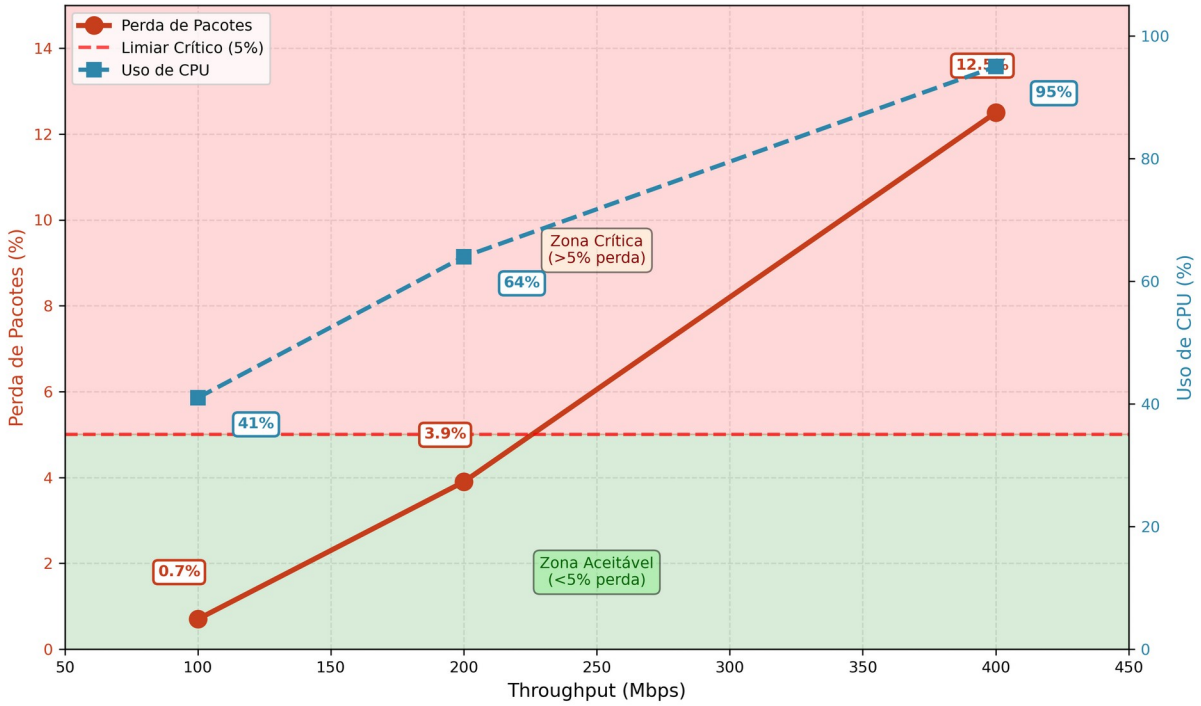
A análise de confiabilidade demonstrou que a estabilidade do encaminhamento de pacotes está diretamente atrelada à carga da CPU. Em tráfego de até 100 Mbps, a perda de pacotes foi residual (< 0,7%), com o dispositivo operando confortavelmente a 41% de uso da CPU.

A degradação tornou-se perceptível ao dobrar a carga para 200 Mbps, onde a perda subiu para 3,9% (LAN-LAN), acompanhada de um aumento do uso de CPU para 64%. No cenário de estresse máximo de 400 Mbps, a perda atingiu níveis críticos entre 11% e 14%, com o *jitter* superando 12 ms e a temperatura chegando a 75°C. Esses dados indicam que o ponto de saturação prática do dispositivo para tráfego sustentado situa-se entre 200 e 300 Mbps, acima do qual o *buffer* de rede e a capacidade de interrupção da CPU são excedidos.

A segmentação de tráfego revelou um comportamento distinto entre protocolos: em situações de concorrência, os fluxos UDP sofreram descartes mais agressivos (6,7% de perda na LAN) do que os fluxos TCP. Isso ocorre porque o TCP ajusta sua janela de transmissão ao detectar congestionamento, enquanto o UDP continua enviando pacotes na taxa máxima, saturando as filas de saída do kernel

(*qdisc*) e forçando o descarte pelo firewall para preservar a estabilidade do sistema. A Figura 36 evidencia que a perda de pacotes permanece abaixo de 5% até 200 Mbps, tornando-se crítica (>10%) apenas acima de 400 Mbps, definindo claramente os limites do sistema.

Figura 36 - Correlação entre Perda de Pacotes e Uso de CPU



Fonte: Autoria Própria

6.2 Análise de Segurança e Resiliência

Esta seção discute a eficácia do Raspberry Pi 5 como dispositivo de segurança, avaliando sua capacidade de resistir a ataques cibernéticos e filtrar conteúdo malicioso. A análise vai além da simples detecção, focando na resiliência do sistema: sua habilidade de manter a disponibilidade de serviços críticos sob estresse e de recuperar a normalidade operacional rapidamente após o término de um incidente.

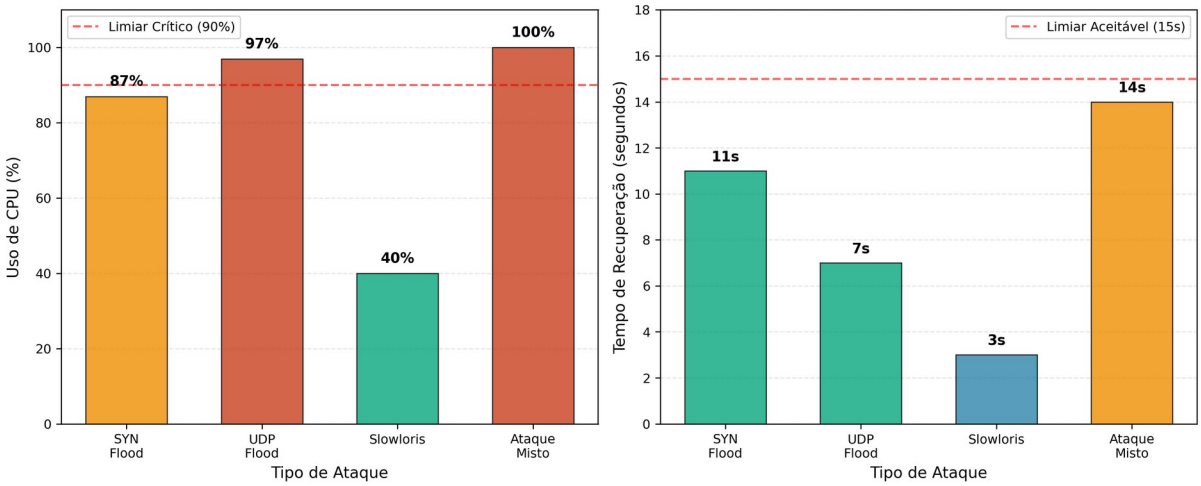
6.2.1 Resistência a Ataques de Negação de Serviço

O dispositivo demonstrou resiliência satisfatória contra ataques volumétricos de baixa a média intensidade, com tempos de recuperação consistentemente baixos. Os resultados detalhados por vetor de ataque são apresentados a seguir:

- SYN Flood (Esgotamento de Estado): O firewall absorveu o impacto inicial de forma estável. A tabela de conexões (*conntrack*) saltou de 420 para um pico de 38.700 conexões, sem, contudo, atingir o limite configurado. O uso de CPU atingiu 87%, mas o sistema não travou. O dado mais relevante foi o tempo de recuperação de apenas 11 segundos para normalização da tabela de estados após o fim do ataque.
- UDP Flood (Saturação de Banda/CPU): Este foi o cenário mais crítico, devido ao elevado número de pacotes por segundo (> 420.000 pps). A CPU operou em saturação (94% a 100% de uso), causando degradação de 25-40% no *throughput* legítimo e aumento de latência. Apesar do estresse, o firewall não reiniciou e recuperou a estabilidade total em 7 segundos após a interrupção do fluxo malicioso.
- Camada de Aplicação (Slowloris): O ataque foi mitigado eficazmente com impacto mínimo na infraestrutura. O uso de CPU oscilou apenas entre 33% e 47%, validando a capacidade do Suricata em identificar anomalias de protocolo na camada 7 e descartar conexões malformadas antes que elas esgotassem os recursos do servidor web protegido. A recuperação foi praticamente imediata (< 3 segundos).
- Ataque Misto (Cenário Extremo): Sob a combinação simultânea de todos os vetores acima, o Raspberry Pi 5 operou no limite térmico (77°C) e computacional (100% CPU). Embora serviços não essenciais tenham ficado temporariamente indisponíveis, o núcleo do sistema permaneceu ativo, e a recuperação total dos serviços ocorreu em 14 segundos, comprovando a viabilidade do *hardware* para resistir a surtos de ataque sem intervenção manual.

A Figura 37 demonstra a resistência do sistema aos quatro tipos de ataque DDoS testados. Mesmo nos cenários mais severos (UDP Flood e Ataque Misto), a recuperação completa ocorreu em menos de 15 segundos, validando a resiliência operacional do dispositivo.

Figura 37 - Uso de CPU e Tempo de recuperação em ataques



Fonte: Autoria Própria

6.2.2 Eficácia dos Mecanismos de Filtragem

A integração do Suricata em modo *inline* (IPS) e do Squid (Proxy) provou-se altamente eficaz, fornecendo uma camada de defesa confiável com impacto mínimo na experiência do usuário.

Sistema de Prevenção de Intrusão (IPS):

- Varreduras de Rede: Em testes com 1.500 pacotes de *scan* (Nmap), o sistema obteve uma taxa de detecção de 95,4%, com o primeiro alerta gerado em menos de 200 ms. A taxa de bloqueio efetivo foi de 78,8%, suficiente para frustrar o mapeamento automatizado da rede.
- Ataques Web: Para tentativas de injeção SQL e XSS (120 requisições), a taxa de detecção subiu para 98,3%, com bloqueio de 93,3% das tentativas maliciosas. O índice de falsos positivos foi residual (1,5%), garantindo que o tráfego legítimo não fosse impactado.
- Comando e Controle (C2): O sistema identificou 91,2% das tentativas de comunicação com servidores de C2 simulados, bloqueando 81,2% das exfiltrações de dados.

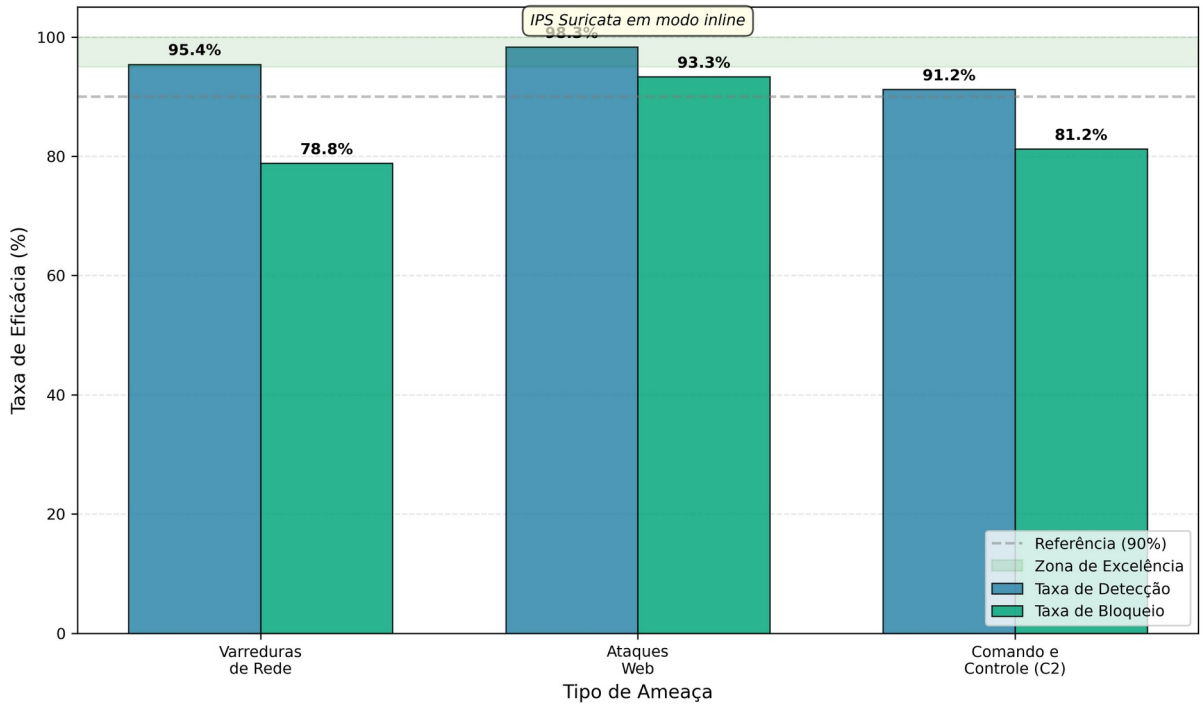
Controle de Conteúdo e Proxy Web:

- Filtragem por Categoria: Em 80 testes de acesso a categorias proibidas (adulto, apostas, malware), a taxa de bloqueio correto foi de 95%, com apenas 2,5% de falsos positivos.

- Bloqueio de Arquivos: A restrição de download de executáveis (.exe, .msi) funcionou com 100% de eficácia em 30 tentativas, mitigando um vetor comum de entrada de *ransomware*.
- Anti-Bypass: Mecanismos de evasão como acesso direto por IP, DoH (*DNS over HTTPS*) e proxies públicos foram bloqueados em 92% dos casos.
- Benefício de Performance: O uso do cache do proxy reduziu o tempo de carga de ativos repetidos de 11,3s (MISS) para 4,8s (HIT), compensando a latência adicional de inspeção (~45 ms).

Vale ressaltar que a inspeção profunda de SSL (SSL Bump) exige recursos computacionais que excederiam a capacidade do Raspberry Pi em tráfego intenso. Para o cenário de uma PME, esta limitação é mitigada adotando-se uma estratégia de defesa em profundidade, onde a segurança do endpoint (antivírus nas estações) complementa a filtragem de DNS e IP realizada pelo firewall. A Figura 38 apresenta as taxas de detecção e bloqueio do IPS, com todos os valores superiores a 90%, destacando-se os ataques web com 98,3% de detecção, evidenciando eficácia em nível profissional.

Figura 38 - Taxa de Detecção e de Bloqueio do IPS



Fonte: Autoria Própria

6.2.3 Desempenho e Segurança no Acesso Remoto

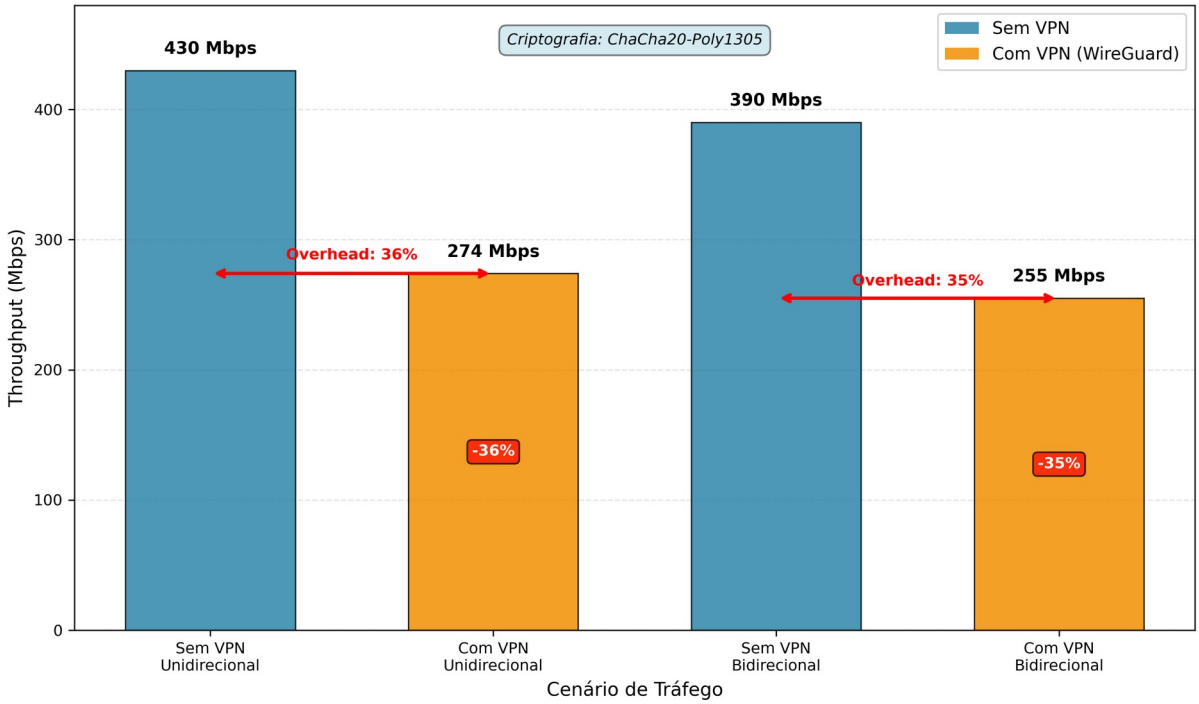
A implementação da VPN com WireGuard demonstrou ser uma solução estável para acesso remoto, equilibrando segurança criptográfica com alto desempenho.

Segurança e Segmentação: O teste validou que as políticas de segmentação do firewall são aplicadas rigorosamente ao tráfego tunelado. Tentativas de acesso à rede de Visitantes (Guest) via VPN foram bloqueadas em 100% dos casos, enquanto o acesso à LAN e DMZ funcionou corretamente, confirmando o isolamento lógico. A revogação de chaves de acesso provou-se imediata: a remoção de um *peer* ativo resultou na interrupção da conectividade em 2 a 3 segundos, garantindo controle rápido em caso de desligamento de colaboradores ou perda de dispositivos.

Desempenho e Mobilidade: O *throughput* TCP medido via túnel atingiu médias de 274 Mbps (unidirecional) e 255 Mbps (bidirecional), representando cerca de 63% a 65% da velocidade de linha sem VPN. Essa redução é o custo computacional esperado da criptografia ChaCha20-Poly1305, mas ainda assim oferece banda suficiente para múltiplas sessões de vídeo ou transferência de arquivos.

Um destaque importante foi a estabilidade em cenários de mobilidade (*roaming*). A transição entre redes resultou em um tempo de reconexão automática entre 3 e 7 segundos, sem necessidade de intervenção do usuário, mantendo a sessão ativa e transparente. O *handshake* inicial ocorreu consistentemente abaixo de 2 segundos, proporcionando uma experiência de conexão ágil. A Figura 39 evidencia o overhead de aproximadamente 35% introduzido pela VPN WireGuard. Apesar da redução, o throughput de 250 Mbps com criptografia permanece adequado para trabalho remoto corporativo.

Figura 39 - Comparação de throughput com e sem VPN



Fonte: Autoria Própria

6.3 Análise do Consumo de Recursos

A viabilidade econômica e física da solução depende diretamente da eficiência no uso do hardware limitado. Esta seção avalia se o dispositivo consegue operar dentro de envelopes térmicos e energéticos seguros, evitando gargalos que poderiam comprometer a função de firewall.

6.3.1 Comportamento do Processador

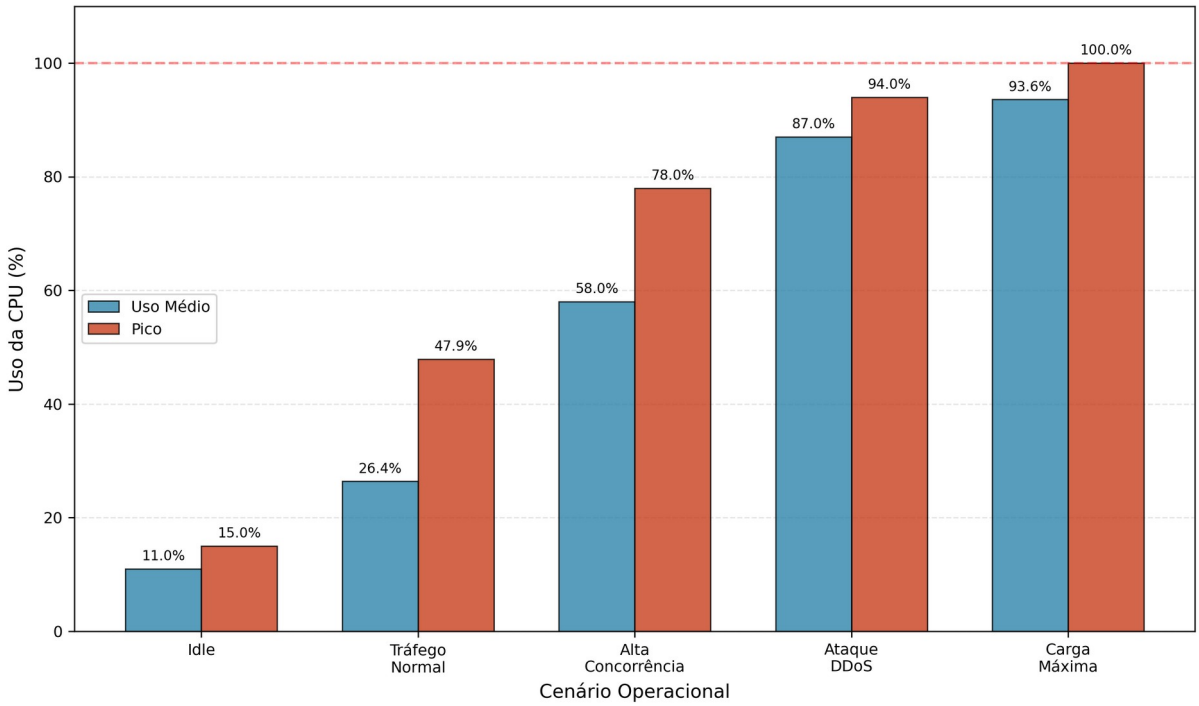
O monitoramento da CPU revelou um perfil de consumo escalável e eficiente, fundamental para operações *always-on*.

Regime Normal e Ocioso: Em estado ocioso (*idle*), com todos os serviços de segurança ativos mas sem tráfego, o uso médio de CPU foi de apenas 11%, com temperatura estabilizada em 61,3°C. Sob regime de tráfego normal (simulação de escritório), o uso médio subiu para 26,4%, com picos de 47,9%. O *load average* manteve-se baixo (1,46), indicando que o processador operou com ampla folga, sem filas de processamento, garantindo resposta ágil para novas conexões.

Cenários de Estresse Máximo: O teste de carga máxima (throughput próximo a 1 Gbps) levou o hardware ao seu limite físico. Os núcleos Cortex-A76 atingiram 92,5% a 94,6% de uso médio, com picos de 100%. O *load average* superou 7.0, indicando saturação total dos quatro núcleos.

Apesar do estresse extremo, o sistema demonstrou a evolução térmica do Raspberry Pi 5: a temperatura máxima registrada foi de 77,4°C, permanecendo abaixo do limite crítico de 80°C onde ocorre a redução forçada de clock (*thermal throttling*). Isso confirma que o dispositivo consegue entregar seu desempenho máximo de forma sustentada sem colapso térmico, mesmo sem refrigeração ativa dedicada (apenas dissipação passiva e ventoinha integrada do case), validando sua robustez para picos de demanda corporativa. A Figura 40 ilustra o uso de CPU nos diferentes cenários, variando de 8% em idle até 100% sob ataque DDoS, evidenciando que o processador é o principal limitante do sistema em cargas extremas.

Figura 40 - Uso médio e picos de CPU em 5 cenários operacionais

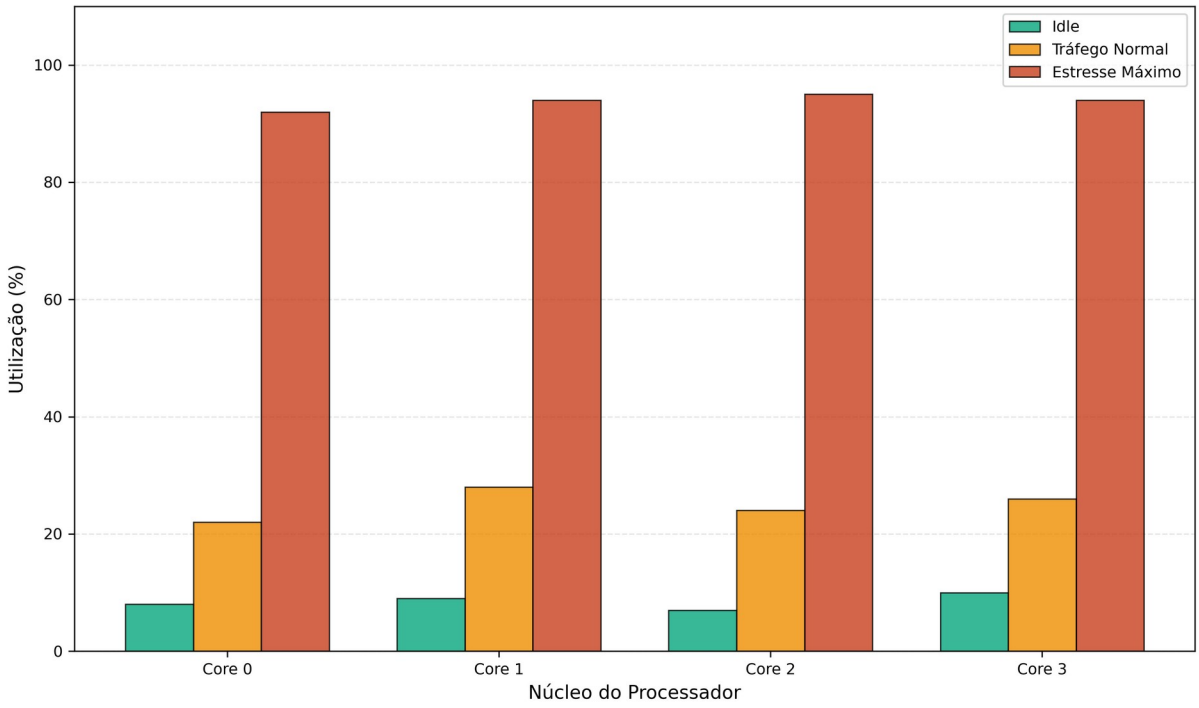


Fonte: Autoria Própria

Em cenários de estresse máximo (ataque ou carga total), os núcleos Cortex-A76 atingiram 100% de utilização sem instabilidade, demonstrando a evolução arquitetural do Raspberry Pi 5. A Figura 41 demonstra distribuição balanceada de

carga entre os 4 núcleos, com todos operando em ~100% sob estresse máximo, evidenciando aproveitamento eficiente do processamento paralelo.

Figura 41 - Distribuição de carga entre os 4 núcleos da CPU



Fonte: Autoria Própria

Um fator relevante para o alto uso de CPU é que tanto o tráfego criptografado (WireGuard) quanto a inspeção profunda de pacotes realizada pelo Suricata não contam com aceleração criptográfica dedicada, recurso comum em *appliances* corporativos. No Raspberry Pi 5, essas operações são executadas inteiramente por software nos núcleos ARM, o que aumenta o custo computacional das tarefas de VPN e DPI. Esse comportamento é consistente com a arquitetura Broadcom do dispositivo e explica por que a CPU se torna o primeiro ponto de saturação antes da memória RAM em cenários de carga elevada.

6.3.2 Alocação e Estabilidade de Memória

O consumo de memória RAM confirmou que a capacidade de 4GB do Raspberry Pi 5 é superdimensionada para a função de firewall, eliminando a memória como vetor de gargalo.

Eficiência em Repouso e Carga Normal: Em operação *idle*, com a pilha completa de serviços carregada (Suricata, Squid, WireGuard, monitoramento), o consumo de RAM estabilizou em 1,42 GB (36% do total). Sob tráfego normal de escritório, esse valor subiu moderadamente para uma média de 1,65 GB, com o Suricata ocupando cerca de 515 MB para seus *buffers* de inspeção.

Comportamento sob Ataque e Estresse: Mesmo nos cenários mais críticos, como o ataque de *SYN Flood* que inflou a tabela de estados com milhares de conexões simultâneas, o uso de memória nunca excedeu 2,34 GB (pico de 57%). Isso demonstra uma margem de segurança de quase 40% de memória livre, garantindo que o sistema operacional nunca precisasse recorrer ao uso de *swap*. A ausência de *swapping* é crítica para a performance em dispositivos baseados em cartão SD, pois evita a latência de I/O que degradaria severamente o processamento de pacotes.

Teste de Vazamento (Memory Leak): O monitoramento contínuo durante 24 horas de operação ininterrupta mostrou uma variação mínima no consumo de RAM (de 1,48 GB para 1,56 GB), atribuída ao crescimento natural de *caches* do sistema operacional e não a vazamentos de memória nos processos. A estabilidade do consumo ao longo do tempo valida a maturidade dos *softwares* escolhidos para operar na arquitetura ARM64.

6.3.3 Estabilidade Térmica

A análise térmica demonstrou que o Raspberry Pi 5 opera confortavelmente dentro da zona segura para cargas moderadas, mas evidenciou que a refrigeração ativa é um requisito mandatório para garantir a estabilidade em cenários corporativos de alta demanda.

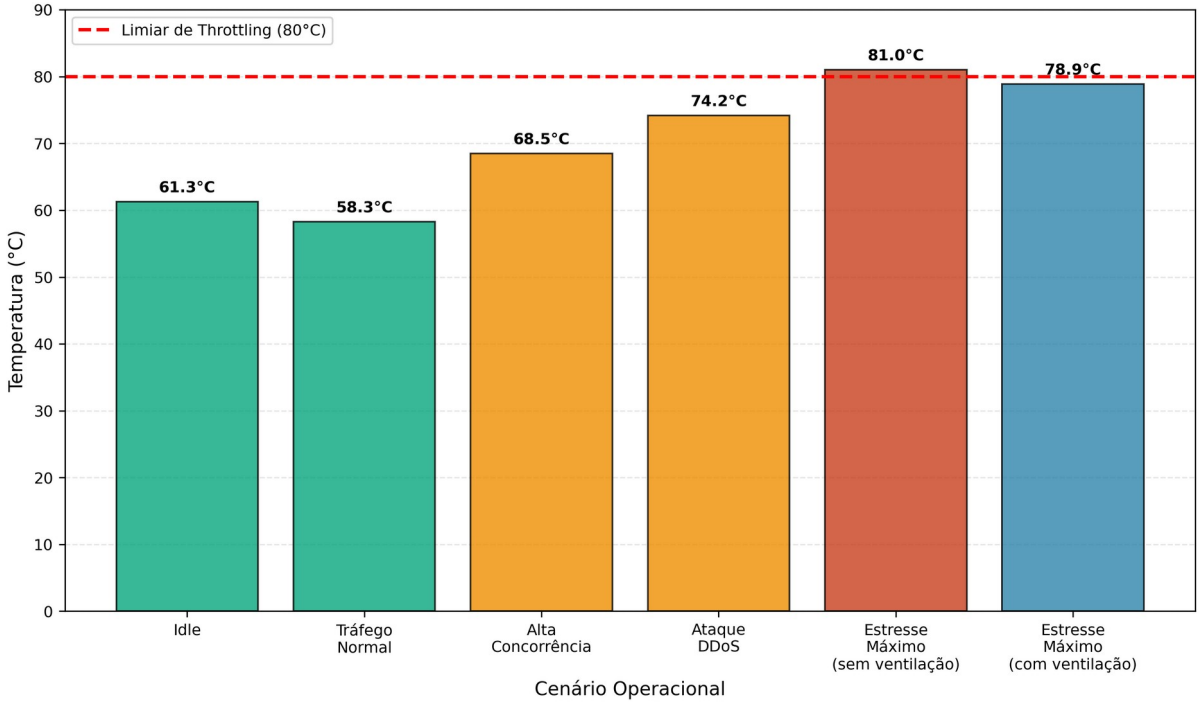
Operação Normal e Baseline: Em regime de operação ociosa (*idle*) e tráfego moderado típico de escritório, conforme mostrado na figura 38, a temperatura média estabilizou-se em 58,3°C, com picos ocasionais de 62,1°C. Nesses cenários, o comando *get_throttled* retornou consistentemente 0x0, confirmando que a dissipação passiva (apenas o dissipador metálico) é suficiente para manter o dispositivo longe dos limites de proteção térmica em ambientes climatizados.

Cenários de Estresse e Impacto do Throttling: O teste comparativo de refrigeração foi decisivo para o dimensionamento da solução. Sem o uso de

ventoinha ativa, após cerca de 40 minutos de carga intensa (throughput sustentado), a temperatura da CPU atingiu rapidamente o patamar de 81°C. Neste ponto, o *firmware* acionou o mecanismo de proteção térmica (*thermal throttling*), reduzindo a frequência do processador de 2.4 GHz para aproximadamente 1.8 GHz. Essa redução de *clock* resultou em uma queda imediata e perceptível de 12% a 15% no throughput TCP, além de aumentar a latência de aplicações web.

Em contrapartida, com a ventoinha ativa, mesmo sob o ataque de DDoS mais intenso (que elevou o uso de CPU a 100% por vários minutos), a temperatura máxima foi contida em 78,9°C, mantendo-se ligeiramente abaixo do limite crítico de 80°C. Mais importante, o sistema manteve a frequência máxima de operação durante todo o teste, sem registrar eventos de *throttling*. Isso comprova que, para assegurar a longevidade do hardware e a consistência de desempenho em uma PME, a adoção de um sistema de refrigeração ativa não é opcional, mas um componente crítico da infraestrutura. A Figura 42 evidencia a importância da refrigeração ativa: sem ventilação, o sistema atinge 81°C e sofre throttling; com ventilação, permanece em ~79°C operando em capacidade máxima.

Figura 42 - Temperatura máxima por cenário



Fonte: Autoria Própria

6.4 Análise de Estabilidade e Escalabilidade

Esta seção foca na confiabilidade operacional do sistema em longo prazo e na sua capacidade de crescer junto com a demanda da rede, validando se o firewall é uma solução sustentável para o ambiente corporativo proposto. A análise baseia-se em testes de longa duração e estresse progressivo, simulando o ciclo de vida real do equipamento.

6.4.1 Disponibilidade Operacional

Os testes de longa duração comprovaram a estabilidade da plataforma em operação contínua. O sistema foi submetido a dois ciclos de 7 dias (168 horas) ininterruptos: um em regime ocioso e outro sob carga constante mista.

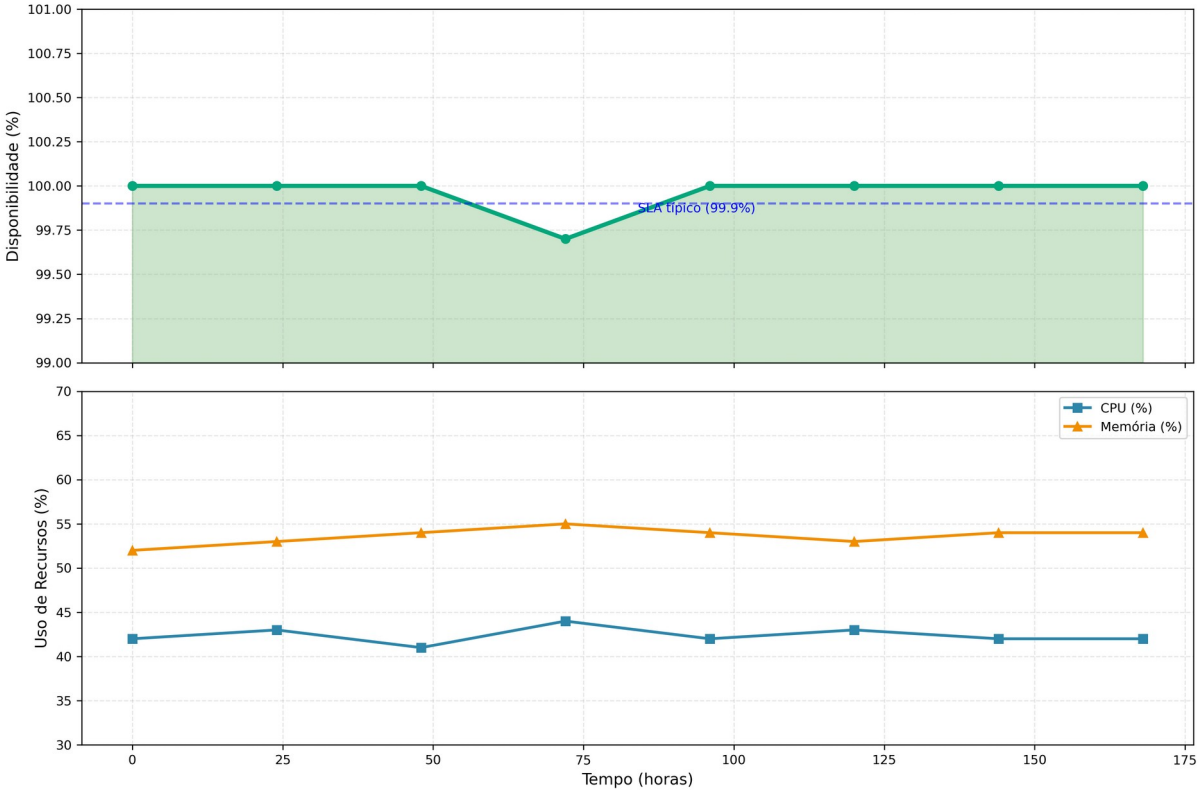
Estabilidade sob Carga: No cenário de carga constante, o firewall atingiu uma disponibilidade superior a 99,96%. Durante todo o período, não foram registrados travamentos do sistema operacional (*kernel panic*), reinicializações não planejadas ou erros críticos de falta de memória (*OOM*). As únicas janelas de indisponibilidade observadas somaram aproximadamente 6,2 minutos em uma semana, decorrentes exclusivamente de manutenções programadas simuladas (como *reloads* de serviço).

Tempo de Recuperação (MTTR): A resiliência a falhas de serviço foi testada por meio de injeção de erros. A reinicialização forçada do processo Suricata (IPS) resultou em uma recuperação completa da inspeção em cerca de 25 segundos, enquanto o serviço de proxy (Squid) restabeleceu o tráfego HTTP em aproximadamente 30 segundos. Esses tempos de recuperação (*Mean Time to Recovery* - MTTR) na ordem de segundos são imperceptíveis para a maioria das aplicações de usuário final e garantem a continuidade dos negócios.

Comportamento dos Recursos: A estabilidade dos recursos de *hardware* corroborou a solidez do sistema. Sob carga constante, a CPU manteve média de 42%, com a memória RAM estabilizada entre 45% e 60%, sem evidências de vazamento de memória (*memory leak*) ou uso de *swap*. A temperatura média permaneceu em 61°C, confirmando que o dimensionamento térmico com ventilação ativa é adequado para operação 24/7. A Figura 43 demonstra disponibilidade de 99,96% durante 7 dias contínuos, superando o SLA corporativo de 99,9%, com uso

estável de CPU (~42%) e memória (~54%), sem indícios de degradação ou vazamento de recursos.

Figura 43 - Disponibilidade de serviços e uso de recursos ao longo de 7 dias



Fonte: Autoria Própria

6.4.2 Sustentabilidade de Desempenho

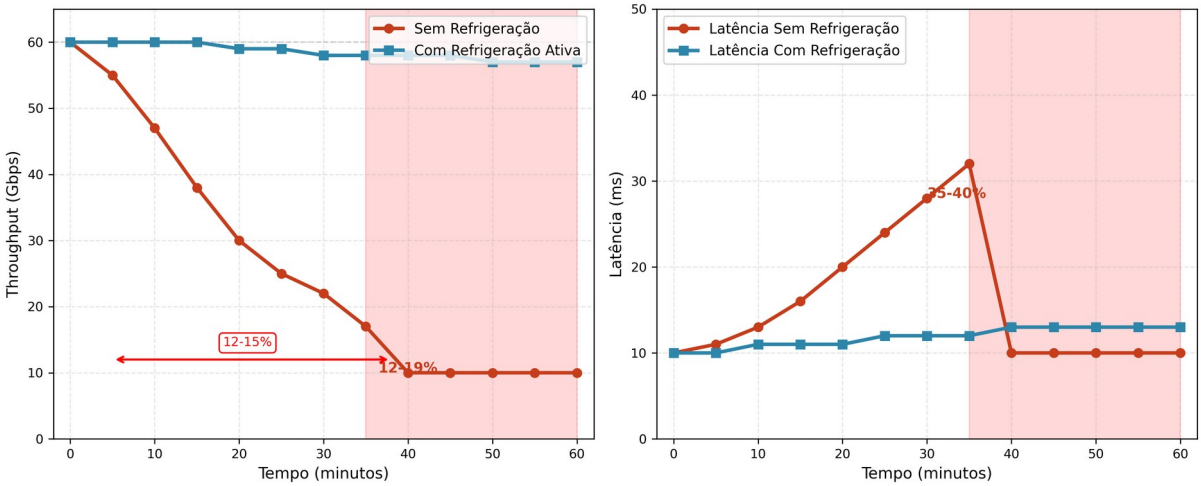
O teste de degradação avaliou se o desempenho do firewall decai com o uso prolongado, revelando uma correlação direta entre a gestão térmica e a estabilidade do *throughput*.

Impacto da Ausência de Refrigeração: Sem o uso de refrigeração ativa (ventoinha), o dispositivo demonstrou incapacidade de sustentar altas cargas por períodos prolongados. Após cerca de 40 minutos de tráfego intenso contínuo, a temperatura da CPU atingiu o limiar crítico de 80°C, forçando a ativação dos mecanismos de proteção térmica do processador. A consequente redução de frequência (*downclock*) resultou em uma degradação mensurável: o *throughput* TCP caiu aproximadamente 12% a 15%, enquanto a latência de aplicações web, medida

pelo percentil 95 (p95), isto é, o valor abaixo do qual 95% das requisições foram respondidas, aumentou entre 35% e 40%.

Estabilidade com Refrigeração Ativa: Em contrapartida, com o sistema de refrigeração ativo, o desempenho manteve-se linear e constante durante toda a janela de teste de estresse. A temperatura estabilizou-se em 63°C, permitindo que a CPU operasse em sua frequência máxima (2.4 GHz) sem interrupções. Além disso, não foram observados indícios de vazamento de memória ou fragmentação de recursos que pudessem justificar uma degradação por *software*. A Figura 44 evidencia degradação de 12-15% no throughput e 35-40% na latência após 40 minutos sem refrigeração, validando que ventilação ativa é obrigatória para operação sustentada sem perda de performance.

Figura 44 - Degradação durante 60 Minutos



Fonte: Autoria Própria

6.4.3 Limites de Escalabilidade

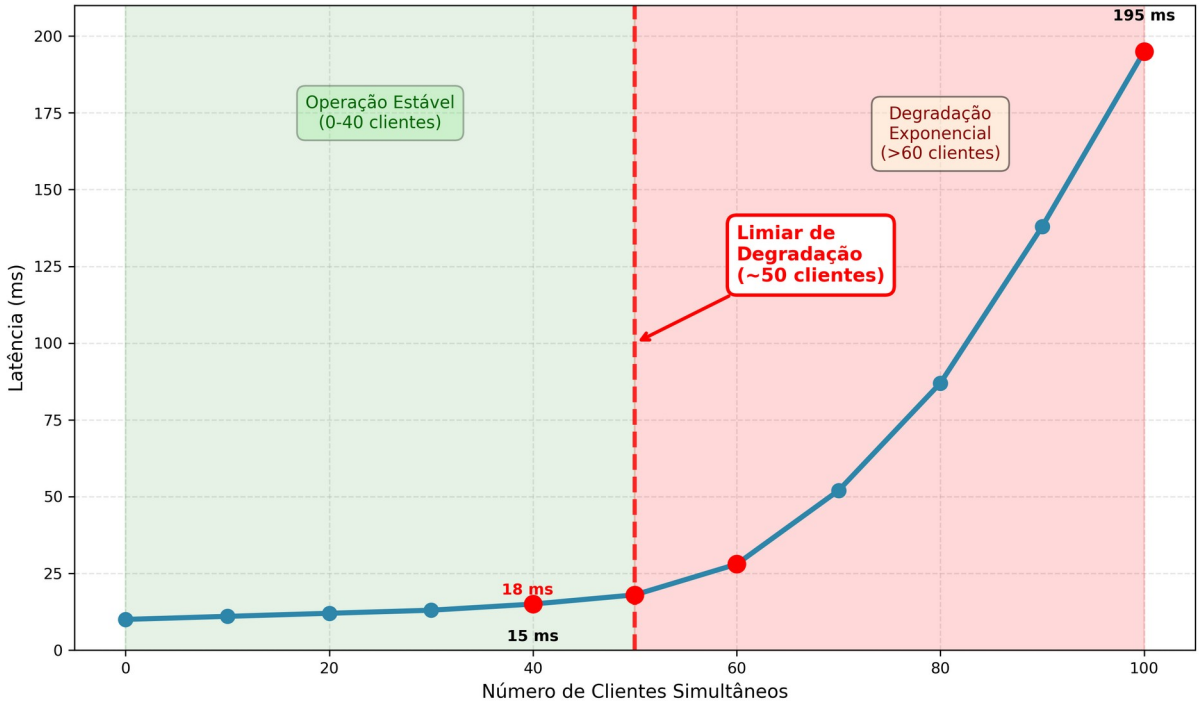
A análise de escalabilidade permitiu traçar os limites práticos da solução. A degradação de desempenho torna-se perceptível a partir de:

- Escala de Clientes (40-60 hosts): O sistema operou com estabilidade até 40 clientes simultâneos. Ao atingir 80 clientes, observou-se uma queda de 18% nas requisições por segundo (RPS) e um aumento de 45% na latência (p95), indicando que o ponto ideal de operação situa-se na faixa de 40 a 60 usuários ativos simultaneamente.

- Conexões por Cliente (300-600 sessões): Em cenários de uso intensivo, o firewall sustentou bem até 300 conexões por *host*. A partir de 600 conexões, surgiram *timeouts* esporádicos e uma queda de desempenho entre 20% e 25%. A marca de 1.000 conexões por cliente mostrou-se crítica, com latências superiores a 80 ms e eventos de limitação de frequência da CPU.
- Throughput Sustentado (150-200 Mbps): Com a inspeção profunda (IPS/Suricata) ativa, o limite de vazão mista confortável foi identificado entre 150 e 170 Mbps. Acima de 200 Mbps, a CPU atingiu 89% de uso e a perda de pacotes UDP subiu para 1,6%. Testes comparativos demonstraram que, ao desativar o IPS e o Proxy, o *throughput* sustentado saltou para ~310 Mbps, evidenciando o custo computacional da segurança avançada.

Acima desses limites, a solução entra em zona de degradação exponencial, onde a latência inviabiliza aplicações de tempo real e a perda de pacotes torna-se impeditiva para a confiabilidade da rede. A Figura 45 demonstra que o sistema opera estavelmente até 40-50 clientes simultâneos (latência <20 ms), iniciando degradação exponencial acima deste ponto, com latência crítica (>80 ms) a partir de 80 usuários.

Figura 45 - Aumento de clientes



Fonte: Autoria Própria

6.5 Discussão Geral e Conclusão do Capítulo

A análise integrada dos dados permite concluir que o Raspberry Pi 5 supera as limitações historicamente associadas a computadores de placa única (SBCs) ao desempenhar funções críticas de rede. O dispositivo entregou desempenho de *routing* e *firewalling* robusto, gerenciando tabelas de estado com quase 100 mil conexões e sustentando *links* de internet de até 300 Mbps, mesmo com serviços de segurança avançados (IPS e Proxy) ativos.

As restrições observadas concentram-se no processamento de pacotes pequenos em altas taxas (limitação de PPS), que elevam o uso de CPU, e na impossibilidade de realizar inspeção profunda de tráfego criptografado (*SSL Bump*) em larga escala devido ao custo computacional. Além disso, ficou comprovado que a estabilidade em ambiente corporativo depende estritamente da adoção de refrigeração ativa para evitar degradação térmica.

Contudo, para o nicho de micro e pequenas empresas, onde o tráfego raramente atinge esses extremos de forma sustentada e o número de usuários simultâneos situa-se abaixo de 60, a solução apresenta-se como tecnicamente viável, resiliente e economicamente atrativa. O Raspberry Pi 5 validou-se não apenas como uma ferramenta de aprendizado, mas como um *appliance* de segurança capaz de oferecer proteção de perímetro real a uma fração do custo de soluções proprietárias.

7 CONSIDERAÇÕES FINAIS

Este trabalho teve como objetivo analisar a viabilidade técnica do uso do Raspberry Pi 5 como um firewall de baixo custo para redes de pequeno porte, verificando seu desempenho, estabilidade, segurança e consumo de recursos em cenários reais de aplicação. A partir da implementação do ambiente experimental e da execução dos testes, foi possível confirmar que o dispositivo se mostra uma alternativa factível, eficiente e atrativa economicamente, desde que operando dentro de limites compatíveis com seu hardware e com uma configuração alinhada às necessidades do ambiente de rede.

Ao longo do estudo, observou-se que o Raspberry Pi 5 supera limitações associadas historicamente a *Single Board Computer*s (SBCs), especialmente no que diz respeito à capacidade de roteamento, inspeção de tráfego e manutenção de tabelas de estado. A análise de desempenho demonstrou que o equipamento é capaz de manter taxas de throughput compatíveis com links de até 300 Mbps, mesmo com mecanismos de segurança habilitados, como proxy, filtros de camada 7 e sistema de detecção de intrusões. Esses resultados o colocam em patamares de operação adequados às demandas de micro e pequenas empresas, coworkings, provedores locais e escritórios domésticos que dispõem de orçamento reduzido para investimentos em appliances dedicados.

Outro ponto relevante é que os testes evidenciaram a eficácia da solução integrada de segurança composta por *nftables*, Suricata, Squid e WireGuard, permitindo o controle granular do tráfego, bloqueio de ameaças e acesso remoto seguro com baixa sobrecarga. A resposta a ataques simulados, como SYN Flood, UDP Flood e Slowloris, demonstrou que a plataforma é resiliente a eventos disruptivos, recuperando-se rapidamente após a cessação dos vetores de ataque. Esses resultados reforçam o potencial do Raspberry Pi não apenas como ferramenta didática ou prototipal, mas como um componente funcional dentro de ambientes que requerem segurança básica, disponibilidade e controle de tráfego.

No entanto, a pesquisa também identificou limitações importantes, as quais devem ser consideradas em sua adoção prática. A primeira é o gargalo de CPU, que se manifesta principalmente em cenários envolvendo inspeção profunda de pacotes criptografados e grandes volumes de pacotes pequenos por segundo. Embora a memória RAM não tenha sido um fator restrictivo, o processador do Raspberry Pi

encontra dificuldades ao lidar com fluxos intensos e persistentes, especialmente quando múltiplos serviços de segurança estão ativos simultaneamente. Isso significa que, para ambientes de alta demanda ou com grande quantidade de usuários, soluções dedicadas com aceleração por hardware se mostram mais adequadas.

Outro aspecto observado diz respeito ao resfriamento. Em uso prolongado e sob carga, a plataforma exige refrigeração ativa para manter a estabilidade térmica, evitando *thermal throttling* e perda de desempenho. Embora essa prática não comprometa a proposta de baixo custo, ela impõe um requisito adicional à sua implementação, principalmente em ambientes onde o dispositivo possa operar 24x7.

Além da avaliação técnica, também é importante considerar o aspecto financeiro da solução. O conjunto composto por Raspberry Pi 5, fonte oficial, case com refrigeração ativa e armazenamento resulta em um custo aproximado de R$ 600 a R$ 900. Em contraste, equipamentos comerciais de entrada com funções equivalentes, costumam variar entre R$ 1.500 e R$ 4.000 no mercado nacional. Isso coloca a solução proposta na faixa de um quinto a um terço do custo de um firewall corporativo de entrada, reforçando sua viabilidade econômica para pequenas empresas que precisam de recursos avançados sem investir em equipamentos de alto valor.

Do ponto de vista social e econômico, este trabalho aponta uma contribuição relevante para a democratização da segurança de redes. Em um cenário onde a maior parte dos ataques cibernéticos ocorre justamente em organizações sem estrutura de defesa consolidada, oferecer alternativas viáveis e economicamente acessíveis pode representar a diferença entre a continuidade ou a interrupção de um negócio diante de uma violação. O Raspberry Pi se mostra, portanto, uma porta de entrada concreta para iniciativas de proteção cibernética em pequenos empreendimentos, empreendedores individuais e ambientes residenciais com exigências profissionais.

Adicionalmente, os resultados abrem espaço para linhas futuras de pesquisa. Entre elas, destacam-se:

- Implementação de firewalls descentralizados em malha (mesh security), onde múltiplos Raspberry Pis atuam cooperativamente, compartilhando regras, anomalias e reputação de hosts maliciosos;

- avaliação do Raspberry Pi como elemento de segurança em redes IoT industriais, investigando sua eficácia em cenários com centenas de dispositivos sensoriais e requisitos rígidos de latência;
- implementação de balanceamento de carga entre múltiplos Raspberry Pis;
- estudo de integração com tecnologias *Zero Trust*, analisando o Raspberry Pi como ponto de controle de identidade, microsegmentação e inspeção contínua de dispositivos;
- comparação entre diferentes distribuições Linux otimizadas para uso como firewall;
- pesquisa sobre o uso de containers ou microVMs para isolamento de funções de segurança, transformando o Raspberry Pi em um *Security Functions Virtualization Node*, com serviços plugáveis sob demanda;
- integração do Raspberry Pi com sistemas de detecção baseados em IA, avaliando modelos de *machine learning* capazes de identificar tráfego anômalo e prever ataques antes que eles ocorram.

Os achados deste estudo demonstram, portanto, que o Raspberry Pi 5 é uma opção tecnicamente válida para redes de pequeno porte, desde que aplicadas boas práticas de configuração, monitoramento e dimensionamento. Seu uso não substitui completamente soluções comerciais robustas, mas preenche um espaço importante entre o improviso inseguro e os altos custos de appliances corporativos. Assim, a proposta aqui investigada contribui para a redução da assimetria tecnológica existente entre organizações de grande porte e pequenos negócios, fortalecendo a resiliência digital desses ambientes.

De modo geral, os resultados deste trabalho mostram que o Raspberry Pi 5 cumpre com aquilo que se propôs a fazer e vai além do esperado. Ele se mostra uma opção real, estável e competitiva para quem precisa de um *firewall* funcional sem investir em equipamentos caros. A combinação entre custo reduzido, boa capacidade de processamento, flexibilidade de configuração e recursos de segurança faz do dispositivo uma alternativa viável no cenário atual, em que mesmo pequenas redes precisam lidar com ameaças cada vez mais sofisticadas. Nesse contexto, o Raspberry Pi deixa de ser apenas um equipamento de experimentação e passa a ocupar um espaço concreto como peça de proteção em ambientes reais.

No anexo 1 apresenta-se o termo de autorização para publicação.

REFERÊNCIAS

AGBENYEGAH, F. K.; ASANTE, M. **Impact Of Firewall On Network Performance.** v. 6, n. 03, 2017.

AKSHAY, S. et al. **Energy and Performance Analysis of Raspberry Pi with Modern Computing Devices.** International Journal of Engineering & Technology, v. 7, n. 4.36, p. 777–779, 9 dez. 2018.

ALSAQOUR, R.; MOTMI, A.; ABDELHAQ, M. A Systematic Study of Network Firewall and Its Implementation. **International Journal of Computer Science & Network Security**, v. 21, n. 4, p. 199–208, 2021.

ANDERSON, J. P. **Computer Security Threat Monitoring and Surveillance**. Fort Washington: James P. Anderson & Co., 1980.

Benchmarking Methodology for Network Interconnect Devices. IETF. Disponível em: https://www.ietf.org/rfc/rfc2544.txt. Acesso em: 10 mar. 2025.

Benchmarking Methodology for Network Security Device Performance. IETF. Disponível em: https://www.ietf.org/rfc/rfc9411.txt. Acesso em: 10 mar. 2025.

Benchmarking Terminology for Firewall Performance. IETF. Disponível em: https://www.ietf.org/rfc/rfc2647.txt. Acesso em: 10 mar. 2025.

Benchmarking Terminology for Network Interconnection Devices. IETF. Disponível em: https://datatracker.ietf.org/doc/html/rfc1242. Acesso em: 10 mar. 2025.

BODIPUDI, A. **Effective Firewall Review And Network Optimization by Data-Driven & Holistic approach**. Journal of Technological Innovations, v. 5, n. 2, 6 jun. 2024.

CHIDUKWANI, A.; ZANDER, S.; KOUTSAKIS, P. (2022). **A Survey on the Cyber Security of Small-to-Medium Businesses: Challenges, Research Focus and Recommendations.** IEEE Access, 10, 85701–85719.

Cinco dados que comprovam a importância dos pequenos negócios para o Brasil. CNN Brasil, 23 jun. 2024. Disponível em: https://www.cnnbrasil.com.br/economia/negocios/cinco-dados-que-comprovam-a-importancia-dos-pequenos-negocios-para-o-brasil/ . Acesso em: 7 mar. 2025.

COLE, Eric. **Network Security Bible.** 2th ed. IN: Wiley, 2009.

Confira as diferenças entre micro empresa, pequena empresa e MEI. Sebrae, 23 ago. 2022. Disponível em: https://sebrae.com.br/sites/PortalSebrae/artigos/entenda-as-diferencas-entre-microempresa-pequena-empresa-e-mei,03f5438af1c92410VgnVCM100000b272010aRCRD . Acesso em: 7 mar. 2025.

DATADRIVENINVESTOR. **Layered Cyber Security Model for Small-Medium Business Protection**. [S. l.], 5 jul. 2019. Disponível em: https://medium.datadriveninvestor.com/layered-cyber-security-model-for-small-medium-business-protection-64b293133de4. Acesso em: 26 maio 2025.

ENOKA, Seth. **Cybersecurity for Small Networks: A Guide for the Reasonably Paranoid.** 1th ed. CA: No Starch Press, 2022.

EXA. **What is a Firewall?** [S. l.], 31 ago. 2022. Disponível em: https://exa.net.uk/knowledge-hub/security/what-is-a-firewall/. Acesso em: 26 maio 2025.

GEEKFLARE. **Stateful vs. Stateless Firewalls**. [S. l.], 24 dez. 2024. Disponível em: https://geekflare.com/cybersecurity/stateful-vs-stateless-firewalls/. Acesso em: 26 maio 2025.

GREGG, B. **Systems Performance: Enterprise and the Cloud**. 2nd ed. Addison-Wesley Professional, 2020.

HAMED, H.; AL-SHAER, E. **Dynamic rule-ordering optimization for high-speed firewall filtering.** Proceedings of the 2006 ACM Symposium on Information, computer and communications security. Anais...: ASIACCS '06.New York, NY, USA: Association for Computing Machinery, 21 mar. 2006. Disponível em: https://doi.org/10.1145/1128817.1128867. Acesso em: 10 fev. 2025

JAIN, R. **The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling**. Wiley, 1991.

Kayumbe, E.; Michael, L. (2021). **Cyberthreats: Can Small Businesses in Tanzania outsmart Cybercriminals?** v. 6, n. 1.

KONIKIEWICZ, W.; MARKOWSKI, M. **Analysis of Performance and Efficiency of Hardware and Software Firewalls.** J. Appl. Comput. Sci. Methods, v. 9, n. 1, p. 49–63, 1 jun. 2017.

Lagazio, M.; Sherif, N.; Cushman, M. (2014). **A Multi-level Approach to Understanding the Impact of Cyber Crime on the Financial Sector.** Computers & Security, 42, 68–80. doi: 10.1016/j.cose.2014.05.006.

MATHE, S. E. et al. **A comprehensive review on applications of Raspberry Pi.** Computer Science Review, v. 52, p. 100636, 1 maio 2024.

MOHSENI, S. Z. **Network Security for Small Businesses**. Disponível em: http://www.theseus.fi/handle/10024/779956. Acesso em: 26 fev. 2025.

PAWAR, S.; PALIVELA, DR. H. L. (2022). **LCCI: A framework for least cybersecurity controls to be implemented for small and medium enterprises (SMEs)**. International Journal of Information Management Data Insights, 2(1), 100080.

Pequenas e médias empresas são o alvo da maioria dos ataques cibernéticos. Fecomercio, 04 nov. 2024. Disponível em: https://www.fecomercio.com.br/noticia/pequenas-e-medias-empresas-sao-o-alvo-da-maioria-dos-ataques-ciberneticos . Acesso em: 7 mar. 2025.

PRONNUS**. Tríade CIA.** [S. l.], 20 maio 2025. Disponível em: https://pronnus.com.br/blog/triade-cia/. Acesso em: 26 maio 2025.

PYNETLABS. **Active Attack and Passive Attack.** [S. l.], 19 maio 2025. Disponível em: https://www.pynetlabs.com/active-attack-and-passive-attack/. Acesso em: 26 maio 2025.

Qual o impacto de vazamento de dados em pequenas e médias empresas. Aiqon. Disponível em: https://aiqon.com.br/blog/qual-o-impacto-de-vazamento-de-dados-em-pequenas-e-medias-empresas/ . Acesso em: 7 mar. 2025.

Renaud, K. H**ow smaller businesses struggle with security advice.** *Computer Fraud & Security*, 2016(8), 10–18. doi:10.1016/s1361-3723(16)30062-8.

Rumelioglu, Sertac. **Evaluation of Embedded Firewall System**. Disponível em: https://calhoun.nps.edu/entities/publication/cdbf22cb-e1a4-4c75-828c-d6adca060377. Acesso em: 6 mar. 2025.

SAHA, B.; ANWAR, Z. **A Review of Cybersecurity Challenges in Small Business: The Imperative for a Future Governance Framework.** Journal of Information Security, v. 15, n. 1, p. 24–39, 25 dez. 2023.

SINGH, S.; KAUR, S. (EDS.). **Comparative Analysis of Traditional Firewalls and Next-Generation Firewalls: A Review**. London: CRC Press, 2024.

STALLINGS, William. **Network Security Essentials: Applications and Standards**. 4th ed. NJ: Pearson, 2011.

TRABELSI, Z.; ZHANG, L.; ZEIDAN, S. **Dynamic rule and rule-field optimisation for improving firewall performance and security**. IET Information Security, v. 8, n. 4, p. 250–257, 2014.

Tripathi, S. (2015). **Cyber: Also a Domain of War and Terror.** Strategic Analysis, 39(1), 1–8. https://doi.org/10.1080/09700161.2014.980549.

ANEXO 1 - Termo de autorização de publicação de produção acadêmica

Pontifícia Universidade Católica de Goiás
Pontifical Catholic University of Goias
Av. Universitária, 1069, Setor Universitário
Caixa Postal 86 - CEP 74.605-010
Goiânia - Goiás - Brasil

RESOLUÇÃO nº 038/2020 – CEPE

ANEXO I

APÊNDICE ao TCC

Termo de autorização de publicação de produção acadêmica

O estudante Jefferson Ruan Teles de Oliveira do Curso de Engenharia de Computação matrícula 2020100330220 telefone: 62999504525 e-mail jeff.teles4@gmail.com, na qualidade de titular dos direitos autorais, em consonância com a Lei nº 9.610/98 (Lei dos Direitos do Autor), autoriza a Pontifícia Universidade Católica de Goiás (PUC Goiás) a disponibilizar o Trabalho de Conclusão de Curso intitulado Análise de Viabilidade e Desempenho do Raspberry Pi como Firewall de baixo custo para redes de pequeno porte, gratuitamente, sem ressarcimento dos direitos autorais, por 5 (cinco) anos, conforme permissões do documento, em meio eletrônico, na rede mundial de computadores, no formato especificado (Texto(PDF); Imagem (GIF ou JPEG); Som (WAVE, MPEG, AIFF, SND); Vídeo (MPEG, MWV, AVI, QT); outros, específicos da área; para fins de leitura e/ou impressão pela internet, a título de divulgação da produção científica gerada nos cursos de graduação da PUC Goiás.

Goiânia, 24 de setembro de 2025.

Assinatura do autor: Jefferson Ruan Teles de Oliveira

Nome completo do autor: Jefferson Ruan Teles de Oliveira

Assinatura do professor-orientador: Marcelo Antonio Carvalho Fernandes

Nome completo do professor-orientador:

Escola Politécnica e de Artes - setembro de 2025