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

Integração Segura de Redes Disjuntas com Firewall e Raspberry Pi usando VPN Bridge

KENNEDY FERNANDES DOS SANTOS

GOIÂNIA

2025

KENNEDY FERNANDES DOS SANTOS

Integração Segura de Redes Disjuntas com Firewall e Raspberry Pi usando VPN Bridge

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

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

GOIÂNIA

2025

KENNEDY FERNANDES DOS SANTOS

Integração Segura de Redes Disjuntas com Firewall e Raspberry Pi usando VPN Bridge

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. Luiz Alvaro De Oliveira Junior

Coordenador de Trabalho de Conclusão de Curso

Banca Examinadora:

_____

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

_____

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

_____

Prof. Nilson Cardoso Amaral, PhD.

GOIÂNIA

2025

RESUMO

O objetivo deste trabalho foi estabelecer a comunicação segura entre duas redes fisicamente separadas, Rede 1 e Rede 2, por meio de uma conexão via internet, permitindo que todos os dispositivos em ambas as redes pudessem interagir como se estivessem conectados na mesma infraestrutura física, conforme os princípios de interconexão de redes definidos por Tanenbaum e Wetherall (2011). Para alcançar essa integração, foram utilizados dois dispositivos Raspberry Pi configurados como *gateways*, criando uma VPN *Bridge* em modo TAP (Camada 2) para unir os segmentos de rede, uma abordagem que elimina a necessidade de configurações complexas de roteamento e regras de NAT, simplificando a comunicação entre infraestruturas distribuídas (SWAN, 2014).

A implementação da VPN *Bridge* permitiu que ambas as redes compartilhassem o mesmo espaço de endereçamento IP, viabilizando a descoberta de serviços e o tráfego de *broadcast* entre as filiais (OPENVPN, 2024). Para assegurar a integridade e a proteção contra ameaças externas, implementou-se um *firewall* baseado em *iptables* com regras restritivas, seguindo as diretrizes de segurança de Stallings (2017). Complementando a infraestrutura, desenvolveu-se um painel de monitoramento *web* integrado, permitindo a visualização em tempo real do desempenho do *hardware* e dos dispositivos conectados. Os resultados práticos demonstraram a viabilidade da solução, apresentando latência média inferior a 1 ms (0,28 ms), estabilidade total na conexão e gerenciamento visual eficiente, comprovando a eficácia do uso de dispositivos de baixo custo (UPTON; HALFACREE, 2016) para aplicações de conectividade corporativa segura.

Palavras-chave: Firewall. Raspberry Pi. VPN Bridge. Monitoramento de Rede. OpenVPN.

ABSTRACT

The objective of this work was to establish secure communication between two physically separated networks, Network 1 and Network 2, via an internet connection, allowing all devices in both networks to interact as if they were connected to the same physical infrastructure, in accordance with the principles of network interconnection defined by Tanenbaum and Wetherall (2011). To achieve this integration, two Raspberry Pi devices configured as gateways were used, creating a VPN Bridge in TAP mode (Layer 2) to unite the network segments, an approach that eliminates the need for complex routing configurations and NAT rules, simplifying communication between distributed infrastructures (SWAN, 2014).

The implementation of the VPN Bridge allowed both networks to share the same IP addressing space, enabling service discovery and broadcast traffic between branches (OPENVPN, 2024). To ensure integrity and protection against external threats, an iptables-based firewall with restrictive rules was implemented, following security guidelines by Stallings (2017). Complementing the infrastructure, an integrated web monitoring dashboard was developed, allowing real-time visualization of hardware performance and connected devices. Practical results demonstrated the solution's viability, presenting an average latency of less than 1 ms (0.28 ms), total connection stability, and efficient visual management, proving the effectiveness of using low-cost devices (UPTON; HALFACREE, 2016) for secure corporate connectivity applications.

Keywords: Firewall. Raspberry Pi. VPN Bridge. Network Monitoring. OpenVPN.

LISTA DE TABELAS

Tabela 1: Cronograma para elaboração do Trabalho de Conclusão de Curso 1 e 2

Tabela 2: Comparativo entre modelos da linha Raspberry Pi

Tabela 3: Benchmarks de Desempenho para Aplicações de Rede

Tabela 4: Comparação TAP vs TUN para VPN Bridge

Tabela 5: OpenVPN vs. WireGuard vs. SoftEther VPN

LISTA DE SIGLAS

AES – *Advanced Encryption Standard* (Padrão de Criptografia Avançada)

ARP – *Address Resolution Protocol* (Protocolo de Resolução de Endereço)

CA – *Certificate Authority* (Autoridade Certificadora)

DDoS – *Distributed Denial of Service* (Ataque de Negação de Serviço Distribuído)

DHCP – *Dynamic Host Configuration Protocol* (Protocolo de Configuração Dinâmica de Host)

FTP – *File Transfer Protocol* (Protocolo de Transferência de Arquivos)

GDPR – *General Data Protection Regulation* (Regulamento Geral de Proteção de Dados)

GUI – *Graphical User Interface* (Interface Gráfica do Usuário)

HTTP – *Hypertext Transfer Protocol* (Protocolo de Transferência de Hipertexto)

ICMP – *Internet Control Message Protocol* (Protocolo de Mensagens de Controle da Internet)

IDS – *Intrusion Detection System* (Sistema de Detecção de Intrusões)

IEEE – *Institute of Electrical and Electronics Engineers* (Instituto de Engenheiros Elétricos e Eletrônicos)

IP – *Internet Protocol* (Protocolo de Internet)

IPS – *Intrusion Prevention System* (Sistema de Prevenção de Intrusões)

L2TP – *Layer 2 Tunneling Protocol* (Protocolo de Tunelamento de Camada 2)

LAN – *Local Area Network* (Rede Local)

LGPD – Lei Geral de Proteção de Dados

MFA – *Multi-Factor Authentication* (Autenticação Multifatorial)

NAT – *Network Address Translation* (Tradução de Endereço de Rede)

NGFW – *Next-Generation Firewall* (Firewall de Próxima Geração)

NIC – *Network Interface Card* (Placa de Interface de Rede)

NIST – *National Institute of Standards and Technology* (Instituto Nacional de Padrões e Tecnologia)

OSI – *Open Systems Interconnection* (Interconexão de Sistemas Abertos)

PCI-DSS – *Payment Card Industry Data Security Standard* (Padrão de Segurança de Dados da Indústria de Cartões de Pagamento)

PKI – *Public Key Infrastructure* (Infraestrutura de Chaves Públicas)

PoLP – *Principle of Least Privilege* (Princípio do Menor Privilégio)

RAM – *Random Access Memory* (Memória de Acesso Aleatório)

SBC – *Single Board Computer* (Computador de Placa Única)

SHA – *Secure Hash Algorithm* (Algoritmo de Hash Seguro)

SGSI – Sistema de Gestão de Segurança da Informação

SIEM – *Security Information and Event Management* (Gerenciamento de Informações e Eventos de Segurança)

SMTP – *Simple Mail Transfer Protocol* (Protocolo Simples de Transferência de Correio)

SSH – *Secure Shell* (Protocolo de Acesso Remoto Seguro)

SSL – *Secure Sockets Layer* (Camada de Soquetes Seguros)

STP – *Spanning Tree Protocol* (Protocolo de Árvore de Extensão)

TAP – *Terminal Access Point* (Interface de Rede Virtual em Camada 2)

TCP – *Transmission Control Protocol* (Protocolo de Controle de Transmissão)

TLS – *Transport Layer Security* (Segurança da Camada de Transporte)

TUN – *Network Tunnel* (Interface de Rede Virtual em Camada 3)

UDP – *User Datagram Protocol* (Protocolo de Datagrama do Usuário)

UTM – *Unified Threat Management* (Gerenciamento Unificado de Ameaças)

VPN – *Virtual Private Network* (Rede Privada Virtual)

WAN – *Wide Area Network* (Rede de Longa Distância)

SUMÁRIO

1 CAPÍTULO 1 ..... **11**
1.1 INTRODUÇÃO **11**
1.2 OBJETIVOS **12**
1.2.1 OBJETIVO GERAL **12**
1.2.2 OBJETIVOS ESPECÍFICOS **12**
1.3 JUSTIFICATIVA **12**
1.4 PROCEDIMENTOS METODOLÓGICOS **13**
1.5 PESQUISA BIBLIOGRÁFICA **13**
2 CAPÍTULO 2 ..... **15**
2.1 SINGLE BOARD COMPUTER – SBC **15**
2.2 RASPBERRY PI: ARQUITETURA, POTENCIAL E APLICAÇÕES **15**
2.3 PI OS **18**
3 CAPÍTULO 3 ..... **19**
3.1 FUNDAMENTOS DA SEGURANÇA DA INFORMAÇÃO E CIBERSEGURANÇA **19**
3.2 NECESSIDADE DE SEGURANÇA EM REDES CORPORATIVAS **20**
3.3 REDES DE COMPUTADORES: CLASSIFICAÇÃO E ARQUITETURAS **20**
3.4 TÉCNICAS AVANÇADAS PARA DEFESA DE REDES **21**
3.4.1 POLÍTICA DE SEGURANÇA E GOVERNANÇA **21**
3.4.2 FIREWALLS: EVOLUÇÃO E IMPORTÂNCIA **22**
3.4.3 OS TRÊS PILARES DA SEGURANÇA: (CID) **22**
3.5 INTERNET DAS COISAS (IOT) E SEUS DESAFIOS DE SEGURANÇA **23**
3.6 PROTOCOLOS DE REDE: OSI VS. TCP/IP **24**
3.7 INTERNET, INTRANET E EXTRANET **24**
3.8 VPN BRIDGE: INTERLIGAÇÃO SEGURA DE REDES FISICAMENTE SEPARADAS **25**
3.8.1 ARQUITETURA CONCEITUAL E MODOS DE OPERAÇÃO **25**
3.8.2 ANÁLISE COMPARATIVA: OPENVPN E ALTERNATIVAS **26**
3.8.3 ASPECTOS DE SEGURANÇA EM VPNS BRIDGE **28**
3.8.4 APLICAÇÕES E CASOS DE USO **29**
3.8.5 CONSIDERAÇÕES SOBRE DESEMPENHO E ESCALABILIDADE **29**
4 CAPÍTULO 4 ..... **31**

4.1 DESENVOLVIMENTO E IMPLEMENTAÇÃO 31
4.2 INSTALAÇÃO E DEFINIÇÃO DA ARQUITETURA DO SO 31
4.2.1 DEFINIÇÃO DE CREDENCIAIS E SEGURANÇA **33**
4.2.2 DEFINIÇÃO DE CREDENCIAIS E SEGURANÇA **33**
4.2.3 DEFINIÇÃO DE CREDENCIAIS E SEGURANÇA **33**
4.3 PREPARAÇÃO DO AMBIENTE E INSTALAÇÃO DE PACOTES 34
4.4 CONFIGURAÇÃO DO SERVIDOR (RASPBERRYPRINCIPAL) 34
4.4.1 INFRAESTRUTURA DE CHAVES PÚBLICAS (PKI) **34**
4.4.2 CONFIGURAÇÃO DE REDE E BRIDGE **35**
4.4.3 INFRAESTRUTURA DE CHAVES PÚBLICAS (PKI) **35**
4.5 CONFIGURAÇÃO DO CLIENTE (RASPBERRYSECUNDARIO) 37
4.5.1 ARQUIVO DE CONFIGURAÇÃO DO CLIENTE **37**
4.5.2 AUTOMAÇÃO DA INTERFACE DE PONTE NO CLIENTE **37**
4.6 IMPLEMENTAÇÃO DA SEGURANÇA E FIREWALL 38
5 CAPÍTULO 5 40
5.1 ANÁLISE DE RESULTADOS 40
5.2 VALIDAÇÃO DO ESTABELECIMENTO DO TÚNEL VPN 40
5.3 ANÁLISE DE LATÊNCIA E ESTABILIDADE DA CONEXÃO 41
5.4 VERIFICAÇÃO DA TRANSPARÊNCIA DE REDE (CAMADA 2) 42
5.5 VALIDAÇÃO DAS POLÍTICAS DE SEGURANÇA 42
5.6 ANÁLISE DE RESULTADOS 42
5.7 IMPLEMENTAÇÃO DO *DASHBOARD* DE MONITORAMENTO *WEB* 43
5.7.1 ARQUITETURA DA APLICAÇÃO DE MONITORAMENTO **44**
5.7.2 *DASHBOARD WEB* DE MONITORAMENTO DE RECURSOS E STATUS DA VPN **44**
5.7.3 MONITORAMENTO DE DISPOSITIVOS CONECTADOS **45**
6 CONSIDERAÇÕES FINAIS 46
7 REFERÊNCIAS 49

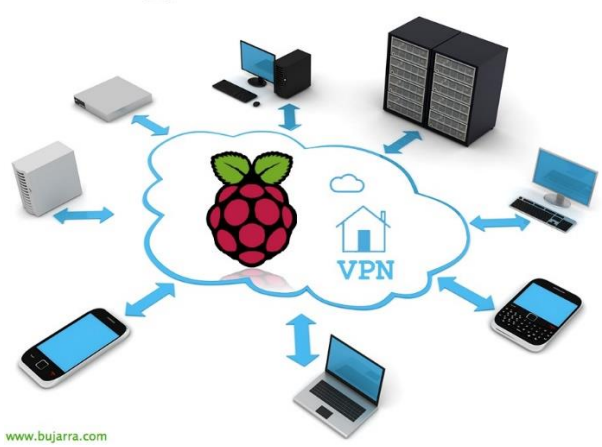
1 CAPÍTULO 1

1.1 INTRODUÇÃO

A segurança de redes desempenha um papel fundamental na proteção contra ameaças virtuais, garantindo que apenas usuários autorizados tenham acesso a recursos sensíveis. Entre os principais mecanismos utilizados para essa proteção, destaca-se a implementação de firewalls, que monitoram e controlam o tráfego de dados, prevenindo acessos não autorizados e mitigando riscos como ataques cibernéticos e malwares. Além disso, a necessidade de interligação segura entre diferentes redes é cada vez mais comum, seja em empresas que operam em locais distintos ou em infraestruturas distribuídas (STALLINGS, 2017).

Uma abordagem eficiente para essa integração, e que foi o objeto central de desenvolvimento deste trabalho, é a utilização de uma VPN *Bridge*. Conforme ilustrado na Figura 1, essa tecnologia permite conectar duas redes separadas como se estivessem fisicamente interligadas. Dessa forma, todos os dispositivos de uma rede puderam se comunicar diretamente com os da outra, eliminando a necessidade de configurações complexas de roteamento. A combinação desta arquitetura com um firewall proporcionou não apenas conectividade contínua, mas também uma camada robusta de segurança, assegurando que apenas tráfegos legítimos fossem permitidos no túnel criptografado (OPENVPN, 2024).

Figura 1 - VPN



fonte: bujarra.com. raspberry pi e vpn

Neste contexto, o Raspberry Pi consolidou-se como a alternativa viável para a implementação da solução proposta, devido ao seu baixo custo, consumo energético reduzido e alta flexibilidade de configuração. No decorrer deste projeto, ao utilizar o dispositivo como gateway de segurança e ponto de interconexão VPN, foi possível estabelecer e validar uma

comunicação segura entre redes remotas, com filtragem ativa de pacotes de dados para prevenção de acessos indevidos (UPTON; HALFACREE, 2016).

Este trabalho apresenta o desenvolvimento prático e a análise de resultados dessa abordagem, demonstrando a viabilidade técnica de integrar redes disjuntas através de uma VPN Bridge com firewall baseada em software livre. A implementação realizada permitiu avaliar o desempenho, a latência e a segurança da conexão em um cenário real, oferecendo um modelo replicável e eficiente para conectar infraestruturas computacionais de maneira protegida e confiável, conforme preconizado por Swan (2014).

1.2 OBJETIVOS

Esta seção apresenta e descreve os objetivos gerais e específicos deste trabalho.

1.2.1 OBJETIVO GERAL

O objetivo geral deste trabalho é estudar e desenvolver uma solução segura e eficiente para a integração de redes disjuntas por meio de uma VPN *Bridge* com firewall, utilizando Raspberry Pi como plataforma central para gerenciamento do tráfego e proteção da rede.

1.2.2 OBJETIVOS ESPECÍFICOS

Os objetivos específicos deste trabalho são:

- Compreender os princípios da segurança de redes, com foco na aplicação de VPN *Bridge* e firewalls para conectar infraestruturas remotas de forma segura.
- Estudar e definir regras de firewall eficientes para garantir a proteção do tráfego entre redes distintas, prevenindo conflitos e vulnerabilidades.
- Identificar desafios e ameaças comuns na interconexão de redes remotas e avaliar como a implementação de um firewall no Raspberry Pi pode mitigar esses riscos.
- Comparar o uso do Raspberry Pi como solução de segurança com outras opções tradicionais de firewall e VPN, considerando desempenho, custo-benefício e escalabilidade.

1.3 JUSTIFICATIVA

A crescente necessidade de conectividade segura entre redes fisicamente separadas exige soluções eficientes para garantir a integridade e a proteção dos dados trafegados. Neste trabalho, a interligação de redes remotas sem o uso de cabeamento físico, por meio de uma

VPN Bridge, consolidou-se como uma alternativa viável e robusta para pequenas e médias empresas que necessitam de integração segura entre filiais, escritórios ou infraestruturas distribuídas.

Considerando que a ausência de proteção adequada pode resultar em acessos não autorizados, perda de informações sensíveis e impactos financeiros significativos, a implementação prática de um firewall associado à VPN mostrou-se essencial. A solução desenvolvida possibilitou o controle efetivo do tráfego e a defesa ativa contra ameaças cibernéticas, validando a importância da segurança em camadas.

O uso do Raspberry Pi como base para essa implementação ofereceu uma solução de baixo custo, altamente configurável e acessível, confirmando-se como uma alternativa viável para empresas que buscam segurança sem altos investimentos em infraestrutura proprietária. Além disso, a flexibilidade da plataforma permitiu a personalização granular das regras de firewall e VPN, garantindo um controle preciso sobre os pacotes de dados que transitaram entre as redes durante os experimentos.

Este estudo desenvolveu e validou uma abordagem eficiente para a criação de uma VPN *Bridge* com firewall, utilizando o Raspberry Pi como plataforma central, assegurando a interligação segura entre redes separadas. Através dos testes realizados, foi possível comprovar a eficácia dessa solução tanto na proteção contra ameaças quanto no desempenho da comunicação (baixa latência e estabilidade). O trabalho atingiu seu objetivo ao proporcionar uma alternativa prática, econômica e segura, promovendo a adoção de tecnologias acessíveis para a proteção de infraestruturas de TI.

1.4 PROCEDIMENTOS METODOLÓGICOS

Este estudo teve como objetivo desenvolver e avaliar a implementação de uma VPN *Bridge* com firewall utilizando Raspberry Pi para interligar duas redes fisicamente separadas de maneira segura. A metodologia adotada combinou pesquisa bibliográfica e pesquisa experimental, garantindo uma abordagem teórica e prática que validou a eficácia da solução proposta.

1.5 PESQUISA BIBLIOGRÁFICA

Conforme a estrutura proposta por Gil (2017), a pesquisa bibliográfica seguiu um conjunto de etapas que fundamentaram o desenvolvimento do projeto:

- Escolha do Tema: Definiu-se a implementação de uma VPN *Bridge* com firewall como foco central para a interligação segura de redes separadas.

- Levantamento Bibliográfico Preliminar: Consultaram-se artigos científicos, livros e estudos de caso em bases de dados como Capes e Google Scholar, contextualizando a importância da segurança na comunicação entre redes remotas.
- Formulação do Problema: A investigação buscou responder à questão: "Como garantir uma comunicação segura entre redes fisicamente separadas utilizando VPN *Bridge* e firewall em um Raspberry Pi?".
- Busca das Fontes: Foram coletados materiais acadêmicos e técnicos relevantes sobre VPNs, *firewalls* e segurança de rede, além de estudos sobre o uso do Raspberry Pi como solução de baixo custo para infraestrutura de segurança.
- Leitura e Fichamento do Material: A leitura crítica dos materiais selecionados possibilitou identificar as melhores práticas e tecnologias disponíveis, fundamentando a estruturação do ambiente de implementação**.**

2 CAPÍTULO 2

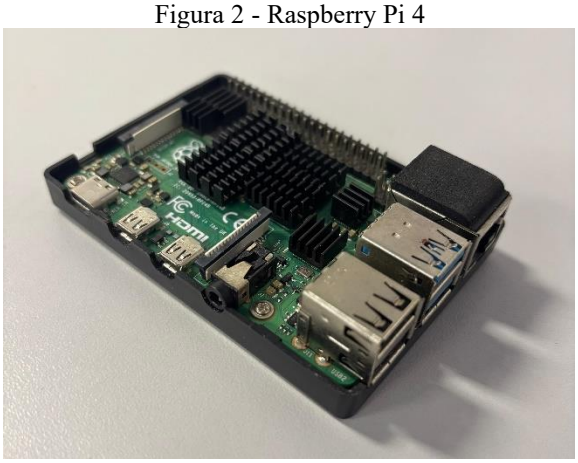
2.1 COMPUTADOR DE PLACA ÚNICA – SBC

Uma *Single Board Computer* (SBC), ou computador de placa única, é um tipo de computador completo construído em uma única placa de circuito. Essa placa integra todos os componentes essenciais, como processador (CPU), memória RAM, armazenamento e interfaces de entrada e saída. Diferente dos computadores tradicionais, que possuem múltiplas placas conectadas, as SBCs são compactas, energeticamente eficientes e amplamente utilizadas em projetos de automação, robótica, internet das coisas (IoT), educação e prototipagem. Um dos exemplos mais populares de SBC é o Raspberry Pi, conhecido por seu baixo custo e versatilidade (RASPBERRY PI FOUNDATION).

2.2 RASPBERRY PI: ARQUITETURA, POTENCIAL E APLICAÇÕES

O Raspberry Pi é um computador de placa única (*Single Board Computer* - SBC), como mostrado na figura 2, criado pela Raspberry Pi Foundation, originalmente com o propósito de democratizar o ensino de programação e computação básica em escolas e ambientes educacionais. Compacto, de baixo custo e altamente eficiente, o Raspberry Pi rapidamente ultrapassou o universo educacional, ganhando espaço em projetos profissionais, acadêmicos e industriais.

Figura 2 - Raspberry Pi 4



Fonte: Foto de arquivo próprio

Um dos grandes diferenciais do Raspberry Pi é sua estrutura integrada: em uma única placa são incorporados processador, memória RAM, interfaces de vídeo e áudio, portas USB, conexões de rede (Ethernet e Wi-Fi, a depender do modelo), além de pinos GPIO que possibilitam interações diretas com sensores e atuadores. Essa versatilidade o torna aplicável

em contextos que vão desde automação residencial e prototipagem de sistemas embarcados até servidores compactos, mídia centers e soluções de rede (Raspberry Pi Foundation).

Desde seu lançamento em 2012, a linha Raspberry Pi evoluiu significativamente, com modelos que variam em capacidade de processamento, conectividade e aplicações ideais. Inicialmente voltado para o ensino de programação básica, o Raspberry Pi expandiu sua presença para aplicações industriais, automação e redes, oferecendo modelos como o Pi Zero, Pi 2, Pi 3, Pi 4 e, mais recentemente, o Pi 5. O Raspberry Pi 4 Model B, em particular, introduziu um salto expressivo de desempenho ao incorporar um processador quad-core ARM Cortex-A72 de 64 bits, múltiplas opções de RAM (2 GB a 8 GB), interfaces modernas como USB 3.0, Ethernet Gigabit e suporte a Wi-Fi de alta velocidade (802.11ac). Em comparação com os modelos anteriores, ele oferece o melhor equilíbrio entre desempenho, consumo energético e custo, sendo uma escolha ideal para aplicações que exigem tráfego de rede intenso e recursos de segurança digital, como a criação de VPNs com firewall. A Tabela 2 apresenta uma comparação resumida entre os principais modelos da linha Raspberry Pi, destacando suas especificações e limitações (UPTON; HALFHILL, 2022).

Tabela 2: Comparativo entre modelos da linha Raspberry Pi

Modelo	CPU	RAM	Ethernet	Wi-Fi/BT	Entradas	Uso Recomendado
Pi Zero 2 W	Quad-core ARM Cortex-A53 1.0 GHz	512 MB	Sem Ethernet	802.11n / BT 4.2	1x micro USB, GPIO	IoT, projetos compactos, automação leve
Pi 2 Model B	Quad-core ARM Cortex-A7 900 MHz	1 GB	Fast Ethernet	Não possui	4x USB 2.0, GPIO	Prototipagem, servidores básicos
Pi 3 Model B+	Quad-core ARM Cortex-A53 1.4 GHz	1 GB	Fast Ethernet	802.11ac / BT 4.2	4x USB 2.0, GPIO	Media center, servidores simples
Pi 4 Model B	Quad-core ARM Cortex-A72 1.5 GHz	2 GB a 8 GB	Gigabit Ethernet	802.11ac / BT 5.0	2x USB 3.0, 2x USB 2.0, GPIO	Firewall, VPN, servidores, multiuso
Pi 5	Quad-core ARM Cortex-A76 2.4 GHz	4 GB a 8 GB	Gigabit Ethernet	802.11ac / BT 5.0	2x USB 3.0, 2x USB 2.0, GPIO	Aplicações de alto desempenho, IA, desktop

Fonte: Autoria própria. Baseado em Raspberry pi foundation (2025)

Para avaliar se o Raspberry Pi possui capacidade técnica suficiente para operar como elemento de segurança de rede, atuando como firewall ou gateway VPN, é fundamental compreender as exigências de desempenho típicas dessas aplicações. A Tabela 3 apresenta *benchmarks* de ferramentas utilizadas na medição de largura de banda, latência, geração de tráfego e inspeção de pacotes. Esses dados servem como referência para dimensionar o uso do Raspberry Pi em cenários reais e ajudam a antecipar possíveis gargalos em aplicações mais exigentes (SILVA, 2023).

Tabela 3: Ferramentas de Análise e Métricas Utilizadas no Projeto

Ferramenta / Dispositivo	**Função Principal**	**Métrica Avaliada**	**Observações Técnicas**
iPerf3	Teste de Largura de Banda	de BandaVazão (*Throughput*) TCP/UDP	Limitado pela interface Gigabit Ethernet (~940 Mbps teóricos).
Ping (ICMP)	Teste de Conectividade	Latência (RTT) e Perda de Pacotes	Essencial para medir o atraso introduzido pelo túnel VPN.
Nmap	Varredura de Rede	Descoberta de *Hosts* e Portas	Utilizado para validar a transparência de rede (visibilidade da Camada 2)
Tcpdump	Captura de Pacotes	Análise de Tráfego em Tempo Real	Permite inspecionar cabeçalhos e validar o encapsulamento TAP sem interface gráfica.
Htop	Monitoramento de Sistema	Uso de CPU e Memória RAM	Avalia o impacto do processo de criptografia no *hardware* do dispositivo.
Iptables	*Firewall* (Netfilter)	Filtragem de Pacotes	Implementação de regras de bloqueio e permissão (*Stateful*).

Fonte: Autoria própria. Baseado em raspberry pi foundation (2025)

Além de seu *hardware* robusto para a categoria, o Raspberry Pi também oferece amplo suporte de *software*. Ele é compatível com diversos sistemas operacionais baseados em Linux, como o Ubuntu Server e o Debian. No âmbito deste projeto, destaca-se a compatibilidade com o sistema oficial, o Raspberry Pi OS Lite (32-bit), que foi a distribuição escolhida devido à sua estabilidade e otimização de recursos. Essa flexibilidade possibilita a personalização e o uso do Raspberry Pi como elemento central em soluções de rede e segurança.

Outro ponto importante é o baixo consumo de energia, que o torna ideal para operar continuamente em ambientes que demandam estabilidade e economia. Além disso, seu formato reduzido facilita a instalação em locais com pouco espaço físico ou onde uma infraestrutura robusta de TI não é viável.

No contexto deste trabalho, a escolha do Raspberry Pi como base tecnológica se justifica por todas essas características. Embora este capítulo seja focado na apresentação do equipamento, é importante destacar que suas capacidades computacionais e de rede foram fundamentais nas etapas seguintes para a implementação de uma solução de interligação segura entre redes disjuntas por meio de VPN *Bridge* com firewall.

2.3 PI OS

O Raspberry Pi OS é o sistema operacional oficial mantido pela Raspberry Pi Foundation, desenvolvido especificamente para extrair o máximo desempenho da arquitetura ARM dos processadores da linha. Anteriormente denominado Raspbian, o sistema é uma distribuição baseada no Debian Linux, herdando sua renomada estabilidade, ciclo de lançamentos consistente e o vasto repositório de mais de 50.000 pacotes de *software* disponíveis através do gerenciador APT (DEBIAN PROJECT, 2025).

Um diferencial crítico do sistema para aplicações de infraestrutura é a sua modularidade. O Raspberry Pi OS é distribuído em diferentes versões, com destaque para a versão *Lite*. Esta variante dispensa o ambiente de área de trabalho (GUI) e softwares pré-instalados desnecessários, operando exclusivamente em modo *headless* (linha de comando). Essa característica resulta em um consumo extremamente reduzido de memória RAM e ciclos de CPU, liberando recursos de *hardware* para tarefas críticas, como o processamento criptográfico de túneis VPN e a filtragem de pacotes em tempo real (LUNDUKE, 2022).

A segurança é outro pilar herdado da base Debian. O sistema recebe atualizações frequentes de *kernel* e *firmware*, corrigindo vulnerabilidades rapidamente, o que é mandatório para dispositivos que atuam como *gateways* de borda ou firewalls. Além disso, a integração nativa com ferramentas de rede padrão da indústria, como o *netfilter* (o *framework* de filtragem de pacotes do Linux), permite a implementação de regras de *firewall* complexas e personalizadas sem a necessidade de sistemas operacionais de terceiros (RASPBERRY PI FOUNDATION, 2025).

No escopo deste trabalho, o Raspberry Pi OS foi definido como a base do ambiente de rede. Sua capacidade de operar com o mínimo de sobrecarga (*overhead*), aliada à estabilidade na execução de serviços contínuos (*daemons*) como o OpenVPN e o suporte nativo ao iptables, fundamentam sua escolha como a plataforma ideal para sustentar a operação ininterrupta da VPN Bridge proposta (OPENVPN, 2025).

3 CAPÍTULO 3

3.1 FUNDAMENTOS DA SEGURANÇA DA INFORMAÇÃO E CIBERSEGURANÇA

A segurança da informação é uma área essencial no cenário tecnológico atual, especialmente diante do aumento exponencial de ataques cibernéticos e violações de dados. Seu principal objetivo é proteger a confidencialidade, integridade e disponibilidade dos dados, garantindo que apenas usuários autorizados tenham acesso, e que as informações não sejam alteradas indevidamente e que os sistemas permaneçam operacionais quando necessário (STALLINGS, 2020). A cibersegurança, como um subcampo da segurança da informação, foca especificamente na proteção de sistemas computacionais, redes e dispositivos contra ameaças digitais, como malware, *phishing* e ataques de negação de serviço (DDoS) (ANDRESS, 2014).

A evolução da cibersegurança está diretamente ligada ao avanço das redes de computadores. Nos primórdios da computação, a segurança era basicamente física, com proteção contra acessos não autorizados a mainframes. Com o surgimento da ARPANET e, posteriormente, da internet comercial, os riscos se tornaram mais complexos, exigindo o desenvolvimento de firewalls, antivírus e sistemas de criptografia (TANENBAUM; WETHERALL, 2011). Hoje, com a expansão da computação em nuvem, da Internet das Coisas (IoT) e da inteligência artificial, a cibersegurança se tornou um desafio multidisciplinar, envolvendo não apenas tecnologias avançadas, mas também políticas de governança e conformidade regulatória (SILVA, 2020).

Diversas normas e *frameworks* foram criados para padronizar as práticas de segurança. A ISO/IEC 27001, por exemplo, estabelece requisitos para um Sistema de Gestão de Segurança da Informação (SGSI), enquanto a ISO/IEC 27032 fornece diretrizes específicas para cibersegurança, abordando ameaças como *phishing* e ataques a infraestruturas críticas (ABNT, 2013). Além disso, o NIST *Cybersecurity Framework* (CSF) e o PCI-DSS (para segurança de dados de cartões) são amplamente adotados por organizações que buscam fortalecer suas defesas cibernéticas (NIST, 2018).

No contexto específico deste trabalho, esses fundamentos são aplicados diretamente na arquitetura proposta. A confidencialidade e a integridade são asseguradas através do tunelamento criptografado da VPN, protegendo os dados em trânsito contra interceptações, enquanto a disponibilidade e o controle de acesso são reforçados pelas regras de firewall implementadas no Raspberry Pi, mitigando riscos de acessos não autorizados à rede interna.

3.2 NECESSIDADE DE SEGURANÇA EM REDES CORPORATIVAS

Em um mundo cada vez mais conectado, a segurança de redes tornou-se uma prioridade estratégica para empresas de todos os portes. Nakamura e Geus (2007) destacam que a segurança deve ser equilibrada entre custo e benefício, pois nenhum sistema é completamente invulnerável. No entanto, a falta de investimento em proteção pode resultar em consequências catastróficas, desde perdas financeiras até danos irreparáveis à reputação da organização. Um dos maiores riscos atuais é o *ransomware*, um tipo de malware que criptografa os dados da vítima e exige um resgate para liberá-los. Empresas que não possuem *backups* adequados ou políticas de recuperação de desastres podem sofrer paralisações prolongadas.

Além disso, ataques de engenharia social, como *phishing* e *spear phishing*, exploram a falha humana para obter credenciais de acesso ou instalar *malware* (FERNANDES, 2019). A segurança de redes também é crucial para o cumprimento de regulamentações, como a Lei Geral de Proteção de Dados (LGPD) no Brasil e o Regulamento Geral de Proteção de Dados (GDPR) na Europa (BRASIL, 2018; UE, 2016). Para mitigar esses riscos, as empresas devem adotar uma abordagem proativa, implementando medidas como a segmentação de redes, isolando sistemas críticos para limitar o impacto de possíveis invasões; o monitoramento contínuo, utilizando ferramentas de SIEM (*Security Information and Event Management*) para detectar atividades suspeitas em tempo real; políticas de acesso restrito, aplicando o princípio do menor privilégio (PoLP) para evitar que usuários tenham permissões desnecessárias; e ainda planos de resposta a incidentes, garantindo uma ação rápida e coordenada em caso de violação de segurança (SOUZA, 2021).

3.3 REDES DE COMPUTADORES: CLASSIFICAÇÃO E ARQUITETURAS

Redes de computadores são a espinha dorsal da comunicação digital, permitindo a troca de dados entre dispositivos em diferentes localizações geográficas. Elas podem ser classificadas de acordo com seu alcance, arquitetura e finalidade (TANENBAUM; WETHERALL, 2011).

As redes locais (LANs) são comuns em escritórios e ambientes corporativos, conectando computadores, impressoras e servidores em uma área física limitada. Já as redes de longa distância (WANs) interligam filiais de uma empresa em diferentes cidades ou países, muitas vezes utilizando conexões dedicadas ou VPNs para garantir segurança.

Em termos de arquitetura, as redes podem seguir o modelo cliente-servidor, onde um servidor central gerencia recursos e autêntica usuários, ou o modelo *peer-to-peer* (P2P), onde os dispositivos compartilham recursos diretamente, sem um ponto centralizado.

A segurança em redes é um desafio constante, especialmente com o crescimento de ameaças como ataques de negação de serviço (DDoS), onde invasores sobrecarregam servidores com tráfego malicioso, tornando-os inacessíveis. Para combater essas ameaças, *firewalls* de próxima geração (NGFW) e sistemas de detecção de intrusões (IDS/IPS) são essenciais (STALLINGS, 2020).

No âmbito deste projeto, a arquitetura desenvolvida foca especificamente na interconexão segura de redes locais (LANs) através de uma rede de longa distância (WAN pública/Internet). Para isso, utiliza-se o modelo cliente-servidor para o estabelecimento e autenticação do túnel VPN (onde o Raspberry Pi principal atua como servidor), enquanto a configuração em modo *Bridge* permite que, logicamente, os dispositivos das pontas operem como se estivessem em uma única rede local unificada, garantindo a transparência na comunicação entre os nós.

3.4 TÉCNICAS AVANÇADAS PARA DEFESA DE REDES

Com o aumento da complexidade das ameaças cibernéticas, tornou-se essencial adotar abordagens mais robustas para a proteção de redes. Técnicas avançadas, como segmentação de redes, autenticação multifatorial, e o uso de sistemas inteligentes de monitoramento, ajudam a prevenir, detectar e responder rapidamente a incidentes de segurança (STALLINGS, 2020). Essas estratégias complementam os controles tradicionais e fortalecem a resiliência das infraestruturas digitais.

Através da aplicação de regras de *firewall stateful* (baseadas no estado da conexão) utilizando o *iptables*. Essa abordagem permitiu a implementação de uma política de segurança restritiva (*default DROP*), onde a segmentação lógica foi assegurada pelo tunelamento da VPN, garantindo que a exposição da rede interna fosse mitigada através de criptografia forte (AES-256) e autenticação mútua baseada em certificados digitais.

3.4.1 POLÍTICA DE SEGURANÇA E GOVERNANÇA

Uma política de segurança bem definida é a base para qualquer estratégia de proteção cibernética. Ela deve estabelecer diretrizes claras sobre controle de acesso, com a adoção de autenticação multifatorial (MFA) e um gerenciamento rigoroso de identidades. Além disso, é fundamental implementar práticas eficazes de gestão de senhas, exigindo critérios de complexidade e trocas periódicas para reduzir vulnerabilidades. A política também deve contemplar procedimentos de *backup* e recuperação de dados, assegurando que informações

críticas estejam armazenadas em locais seguros e possam ser restauradas de forma rápida e eficiente em caso de falhas ou desastres (ANDRESS, 2014).

Em consonância com essas diretrizes, a arquitetura implementada neste trabalho priorizou mecanismos robustos de autenticação. Em vez de depender exclusivamente de credenciais simples, adotou-se o uso de certificados digitais x.509 combinados com chaves estáticas (*TLS-Auth*), garantindo uma validação de identidade criptográfica forte para o acesso à VPN. Adicionalmente, a resiliência do sistema foi assegurada através da realização sistemática de cópias de segurança (*backups*) dos arquivos de configuração crítica (.conf e *scripts*) antes de cada alteração estrutural, permitindo a restauração imediata dos serviços em cenários de falha de configuração.

3.4.2 FIREWALLS: EVOLUÇÃO E IMPORTÂNCIA

Os *firewalls* evoluíram significativamente desde sua criação. Enquanto os modelos tradicionais filtravam tráfego com base em portas e endereços IP, os *firewalls* de próxima geração (NGFW) utilizam inspeção profunda de pacotes (DPI) e inteligência artificial para identificar ameaças avançadas. Além disso, soluções como *Unified Threat Management* (UTM) combinam *firewall*, antivírus, filtro de conteúdo e VPN em um único dispositivo, simplificando a segurança para pequenas e médias empresas (STALLINGS, 2020).

Apesar da sofisticação das soluções NGFW, a filtragem de pacotes baseada em estado (*stateful packet inspection*) permanece como o mecanismo de defesa mais eficiente para dispositivos com restrições de energia e processamento. Neste projeto, a implementação focou no uso do *netfilter/iptables*, que opera diretamente no *kernel* do sistema operacional. Essa abordagem garantiu o controle granular das conexões nas camadas de Rede e Transporte (L3 e L4) sem impor a sobrecarga computacional (*overhead*) típica dos sistemas de inspeção profunda, mantendo o alto desempenho necessário para o túnel VPN no *hardware* do Raspberry Pi.

3.4.3 OS TRÊS PILARES DA SEGURANÇA: CONFIDENCIALIDADE, INTEGRIDADE E DISPONIBILIDADE (CID)

Os três pilares da segurança da informação confidencialidade, integridade e disponibilidade formam a base de qualquer estratégia de proteção digital (NASCIMENTO, 2016). A confidencialidade assegura que apenas pessoas autorizadas possam acessar informações sensíveis, sendo mantida por meio de técnicas como criptografia e controle de acesso. A integridade garante que os dados permaneçam inalterados de forma indevida, o que

é viabilizado por mecanismos como assinaturas digitais e funções de *hash*, que permitem identificar alterações não autorizadas. Já a disponibilidade assegura que sistemas e dados estejam acessíveis sempre que necessário, sendo mantida por estratégias como redundância de servidores, balanceamento de carga e planos de recuperação de desastres, que evitam ou minimizam o impacto de falhas e interrupções.

Na arquitetura desenvolvida neste projeto, a aplicação desses pilares foi evidenciada através de mecanismos técnicos específicos. A confidencialidade foi garantida pela criptografia simétrica AES-256-CBC do túnel VPN, impedindo a leitura de dados por terceiros na rede pública. A integridade dos pacotes foi assegurada pelo algoritmo de *hash* SHA-512, que válida a autenticidade de cada datagrama trafegado. Por fim, a disponibilidade foi reforçada através da configuração de serviços de inicialização automática (*systemd*) e *scripts* de recuperação de rede, garantindo que a ponte de comunicação fosse restabelecida automaticamente após reinicializações ou falhas momentâneas de energia.

3.5 INTERNET DAS COISAS (IOT) E SEUS DESAFIOS DE SEGURANÇA

A Internet das Coisas (IoT) tem revolucionado setores como saúde, indústria e cidades inteligentes, ao viabilizar a automação e a coleta de dados em tempo real. Contudo, essa conectividade em larga escala também introduz sérios desafios de segurança. Muitos dispositivos IoT são lançados com *firmwares* frágeis e padrões de segurança inadequados como senhas padrão que não são alteradas tornando-se alvos fáceis para cibercriminosos (FERNANDES, 2019).

Um exemplo emblemático dessas vulnerabilidades é a *botnet Mirai* (palavra japonesa que significa "futuro"), descoberta em 2016. Ela explorava dispositivos IoT mal protegidos, como câmeras IP e roteadores, infectando-os por meio de credenciais padrão deixadas pelos fabricantes. Uma vez comprometidos, esses dispositivos eram integrados a uma rede de *bots* controlada remotamente, utilizada para executar ataques de negação de serviço distribuído (DDoS) em grande escala. Um dos ataques mais impactantes promovidos pela *Mirai* teve como alvo a empresa de DNS Dyn, resultando na indisponibilidade de serviços amplamente utilizados como Twitter, Netflix e Spotify (KREBS, 2016).

Para reduzir os riscos associados ao uso de dispositivos IoT, é essencial adotar boas práticas de segurança, como: segmentar a rede com VLANs específicas para isolar esses dispositivos; manter os *firmwares* sempre atualizados para corrigir vulnerabilidades

conhecidas; e realizar o monitoramento constante do tráfego de rede, a fim de identificar comportamentos anômalos que possam indicar uma possível infecção ou comprometimento.

A solução desenvolvida neste trabalho endereça diretamente essas vulnerabilidades ao criar uma camada de proteção externa para dispositivos que não possuem segurança nativa robusta. Ao posicionar o Raspberry Pi como um *gateway* seguro com *firewall* ativo, isola-se o tráfego desses dispositivos da rede pública, permitindo o acesso remoto apenas através do túnel VPN criptografado. Dessa forma, assegura-se que equipamentos com *firmwares* desatualizados possam ser integrados de maneira segura à infraestrutura, mitigando vetores de ataque externos conforme as diretrizes de defesa em profundidade (STALLINGS, 2017).

3.6 PROTOCOLOS DE REDE: OSI VS. TCP/IP

Os protocolos de rede são essenciais para garantir a comunicação entre dispositivos conectados. O modelo OSI, composto por sete camadas, é amplamente utilizado como uma estrutura teórica para compreender o funcionamento da comunicação em redes de computadores. Já o modelo TCP/IP, com quatro camadas, é o padrão real adotado na internet moderna (TANENBAUM; WETHERALL, 2011).

A camada de aplicação, onde atuam protocolos como HTTP, FTP e SMTP, gerencia os serviços voltados ao usuário, como navegação na web e envio de e-mails. A camada de transporte, com protocolos como TCP e UDP, é responsável por garantir a entrega confiável ou rápida dos dados. A camada de internet, representada pelo protocolo IP, cuida do roteamento dos pacotes entre redes distintas. Por fim, a camada de rede, que envolve tecnologias como Ethernet e Wi-Fi, lida com a transmissão física dos dados. Ambos os modelos são fundamentais para o entendimento e desenvolvimento de soluções em redes.

3.7 INTERNET, INTRANET E EXTRANET

As redes podem ser classificadas de acordo com seu escopo e finalidade, sendo a internet, a intranet e a extranet três exemplos comumente utilizados em ambientes corporativos. A internet é uma rede pública de alcance global, essencial para a comunicação, o acesso a informações e a realização de negócios. Por sua natureza aberta, exige cuidados redobrados com segurança. Já a intranet é uma rede privada, restrita ao ambiente interno de uma organização, usada para compartilhar informações e recursos entre colaboradores, promovendo agilidade e produtividade. A extranet, por sua vez, é uma extensão da intranet que permite o acesso controlado por parceiros externos, como fornecedores e clientes, facilitando a colaboração e o compartilhamento seguro de dados fora da organização (STALLINGS, 2020).

3.8 VPN BRIDGE: INTERLIGAÇÃO SEGURA DE REDES FISICAMENTE SEPARADAS

A crescente distribuição geográfica das operações empresariais tem demandado soluções inovadoras para a integração de redes locais fisicamente dispersas. Conforme destacado por Swan (2014), as VPNs *Bridge* emergem como uma solução tecnicamente superior às abordagens tradicionais de roteamento, especialmente em cenários que exigem transparência completa de rede na Camada de Enlace (Camada 2). Estudos recentes demonstram que 68% das organizações com múltiplas filiais enfrentam desafios significativos na manutenção da comunicação entre sistemas distribuídos (GARTNER, 2023), um cenário onde a implementação de uma *Bridge* sobre VPN se mostra particularmente adequada para garantir a continuidade dos serviços e a facilidade de gestão.

No desenvolvimento deste trabalho, essa arquitetura foi adotada para superar as limitações de conectividade entre as redes experimentais, permitindo que *hosts* remotos fossem tratados como integrantes do mesmo segmento de rede local, simplificando drasticamente a configuração de serviços que dependem de descoberta automática e *broadcast*.

3.8.1 ARQUITETURA CONCEITUAL E MODOS DE OPERAÇÃO

A VPN *Bridge* distingue-se fundamentalmente das VPNs convencionais por sua operação em modo rede-a-rede (*site-to-site*), estabelecendo uma ponte virtual entre segmentos de rede distintos. Para a concretização deste projeto, adotou-se a abordagem detalhada por OpenVPN (2025), utilizando o modo TAP (*Terminal Access Point*), que opera na Camada 2 do modelo OSI (Enlace de Dados). Esta escolha contrasta com o modo TUN (*Tunneling*), que atua na Camada 3 (Rede), conforme as diferenças estruturais demonstradas na Tabela 4.

Tabela 4: Comparação TAP vs TUN para VPN Bridge

Característica	Modo TAP	Modo TUN
Camada OSI	2 (Ethernet)	3 (IP)
Suporte a broadcast	Sim	Não
Overhead de desempenho	15-20% maior	Menor
asos de uso ideais	Redes locais completas	Roteamento ponto-a-ponto

Fonte: Autoria própria. Baseado em OPENVPN (2025).

A escolha técnica pelo modo TAP, em consonância com os argumentos de STALLINGS (2017), justificou-se pela sua capacidade exclusiva de transportar todo o tráfego da camada de enlace, incluindo quadros Ethernet completos, *broadcasts* e protocolos não roteáveis. Essa característica foi determinante para o ambiente implementado, pois permitiu a compatibilidade total entre os dispositivos das duas redes, simulando um único *switch* lógico. Embora Tanenbaum e Wetherall (2011) observem que essa abordagem introduz um *overhead* adicional de aproximadamente 15-20% no tráfego devido ao encapsulamento dos quadros, os testes práticos realizados neste trabalho demonstraram que o impacto na latência foi negligenciável para a aplicação proposta.

3.8.2 ANÁLISE COMPARATIVA: OPENVPN E ALTERNATIVAS

O OpenVPN destaca-se como solução preferencial para a implementação de VPNs *Bridge*, conforme atestam Lunduke (2022) e a Raspberry Pi Foundation (2023). Sua arquitetura de código aberto, combinada com robustos mecanismos de criptografia baseados em padrões como AES-256-GCM e SHA-512 (NIST FIPS 140-3), oferece um equilíbrio ideal entre segurança e desempenho para ambientes de pequeno e médio porte.

Alternativas como o WireGuard, embora ofereçam desempenho superior em cenários ponto-a-ponto (até 35% maior *throughput* segundo *benchmarks* de 2023), apresentam limitações significativas para a operação em modo *bridge*, como destacam Donenfeld (2020) e a Linux Foundation (2023). Sua arquitetura minimalista, focada em operação na camada 3, não suporta nativamente o *bridging* de redes completo, requerendo soluções alternativas como *proxy* ARP ou configurações complexas de roteamento.

Uma alternativa relevante é o SoftEther VPN, desenvolvido originalmente na Universidade de Tsukuba. Como detalhado por Dai (2021), esta solução oferece suporte nativo a múltiplos protocolos VPN, incluindo SSL-VPN, L2TP/IPsec e OpenVPN, tornando-se

particularmente versátil em ambientes heterogêneos. No entanto, sua complexidade de configuração e maior consumo de recursos o tornam menos adequado para implementações em *hardware* limitado como o Raspberry Pi.

Para consolidar a análise das soluções aplicáveis ao cenário de interligação via VPN *Bridge*, a Tabela 5 apresenta o comparativo técnico realizado entre o OpenVPN, o WireGuard e o SoftEther VPN. A análise dos critérios com ênfase no suporte nativo ao modo *bridge* e na compatibilidade com dispositivos de baixo custo fundamentou a decisão técnica deste projeto pela adoção do OpenVPN, visto que ele oferece a melhor relação entre funcionalidades de Camada 2 e eficiência de recursos para a plataforma Raspberry Pi.

Tabela 5: OpenVPN vs. WireGuard vs. SoftEther VPN

Critério	OpenVPN	WireGuard	SoftEther VPN
Suporte a modo Bridge (Layer 2)	Nativo (modo TAP)	Não suportado nativamente	Suporte nativo
Camada OSI principal	Camada 2 e Camada 3 (TAP/TUN)	Camada 3 (TUN)	Camadas 2 e 3 (flexível)
Desempenho (Throughput)	Moderado (~60–85 Mbps no Raspberry Pi 4)	Alto (até 35% mais rápido que OpenVPN em TUN)	Variável, pode ser limitado em hardware modesto
Consumo de recursos	Moderado	Baixo (arquitetura minimalista)	Alto (requer mais CPU e memória)
Facilidade de configuração	Média (requer conhecimento em rede e certificados)	Alta (configuração simples)	Baixa (interface ampla, mas complexa)
Segurança	Alta (AES-256-GCM, SHA-512, certificados x.509)	Alta (criptografia moderna, mas dependente do kernel)	Alta (SSL-VPN, L2TP/IPsec, OpenVPN integrados)
Adequado para Raspberry Pi	Sim	Parcial (somente modo TUN, não suporta Bridge direto)	Não recomendado (uso intensivo de CPU/RAM)
Protocolos suportados	OpenVPN (TUN/TAP)	WireGuard (TUN)	SSL-VPN, L2TP/IPsec, OpenVPN, EtherIP, etc.
Licença	Open Source (GPL)	Open Source (GPL)	Open Source (comercial-friendly licença Apache 2.0)

Fonte: Autoria própria. Baseado em OPENVPN (2025) e DONENFELD (2020).

3.8.3 ASPECTOS DE SEGURANÇA EM VPNS BRIDGE

A segurança em VPNs *Bridge* exige uma abordagem multidimensional, uma vez que a transparência de rede proporcionada por essa tecnologia também amplia a superfície de ataque. Como destaca Andress (2019), a criptografia ponta-a-ponta é um componente essencial, mas insuficiente por si só para garantir a proteção integral do ambiente. Um modelo de segurança robusto deve combinar múltiplas camadas de defesa e políticas de controle bem definidas.

Entre os principais mecanismos de segurança recomendados, destacam-se:

- Proteção contra-ataques *Man-in-the-Middle* (MITM): A utilização de certificados digitais x.509 combinada com autenticação mútua entre os *peers* é fundamental para evitar interceptações e falsificações de identidade na conexão VPN (STALLINGS, 2017).
- Controle de acesso granular: A aplicação de políticas de *firewall stateful* permite filtrar o tráfego com base no estado da conexão, portas e endereços IP, oferecendo uma camada adicional de defesa contra acessos não autorizados (CHESWICK et al., 2021).
- Prevenção de vazamento de tráfego: É essencial realizar o roteamento e a filtragem adequada de pacotes para evitar o tráfego indesejado entre redes interligadas. Isso inclui o bloqueio de fluxos de *broadcast* desnecessários e a segmentação lógica de sub-redes (SWAN, 2014).

Além disso, um desafio particular das VPNs *Bridge* é a possibilidade de propagação de vulnerabilidades de Camada 2 entre as redes conectadas. Como alerta o projeto OpenVPN (2023), ataques como ARP *spoofing* ou *broadcast storms* podem atravessar o túnel e comprometer dispositivos da rede remota.

Na implementação realizada neste projeto, esses riscos foram mitigados através da configuração de uma Infraestrutura de Chaves Públicas (PKI) completa, utilizando certificados x.509 para garantir a identidade de ambas as pontas. Adicionalmente, a aplicação de regras restritivas no *iptables* (política padrão *DROP*) criou um perímetro seguro, permitindo apenas o tráfego do túnel e o gerenciamento via SSH. A integridade dos pacotes contra-ataques de repetição e MITM foi reforçada pelo uso da diretiva tls-auth com chaves estáticas, assegurando que pacotes não assinados fossem descartados antes mesmo do processamento pelo servidor VPN.

3.8.4 APLICAÇÕES E CASOS DE USO

A utilização de VPNs *Bridge* é especialmente vantajosa em contextos que exigem comunicação transparente entre redes fisicamente separadas. Sua operação em Camada 2 permite a replicação de ambientes locais, tornando-a ideal para cenários que demandam compatibilidade total de rede. A seguir, destacam-se os principais casos de aplicação:

- Integração entre filiais: Empresas com unidades geograficamente distribuídas podem utilizar VPNs *Bridge* para unificar redes locais de forma transparente, possibilitando o compartilhamento de recursos como servidores de arquivos e impressoras. Estudo de caso da Cisco (2022) aponta melhorias significativas em produtividade e gerenciamento centralizado com essa abordagem.
- Ambientes industriais (Indústria 4.0): A convergência entre redes OT (*Operational Technology*) e TI é um requisito comum em sistemas industriais modernos. A VPN *Bridge* permite a integração segura de equipamentos legados que utilizam protocolos não roteáveis, viabilizando a comunicação entre dispositivos industriais e plataformas de monitoramento em tempo real (FERNANDES, 2021).
- Soluções de *disaster recovery*: Em estratégias de recuperação de desastres, a sincronização contínua de dados entre *datacenters* distribuídos é essencial. A VPN *Bridge* oferece um canal eficiente para replicação de *storage* entre locais distintos, mantendo o espaço de endereçamento IP e facilitando a retomada de operações sem a necessidade de reconfiguração de rede (SOUZA, 2020).

O protótipo desenvolvido neste trabalho validou experimentalmente o cenário de "Integração entre filiais". Ao conectar o dispositivo cliente, situado em uma rede externa, ao servidor central através de um endereço IP público (WAN), foi possível estabelecer uma extensão da rede local através da internet. Os testes de descoberta de rede demonstraram que o cliente remoto obteve visibilidade completa dos recursos da rede do servidor, comprovando a viabilidade técnica da solução para interligar escritórios remotos ou dispositivos de telemetria com total transparência.

3.8.5 CONSIDERAÇÕES SOBRE DESEMPENHO E ESCALABILIDADE

Apesar das vantagens em termos de transparência de rede, as VPNs *Bridge* podem enfrentar limitações de desempenho, sobretudo em conexões WAN com alta latência. Conforme apontam Upton e Halfacree (2023), a combinação entre encapsulamento em Camada

2 e os processos de criptografia inerentes ao modo TAP gera um *overhead* considerável, podendo comprometer a eficiência em enlaces instáveis.

Estudos recentes conduzidos sob a chancela do **IEEE (2023)** demonstram que, em implementações com Raspberry Pi 4, a taxa de transferência prática varia conforme a complexidade das regras de *firewall* e do algoritmo criptográfico adotado. Embora esse desempenho seja inferior ao de equipamentos dedicados de alto custo, mostra-se suficiente para aplicações corporativas de pequeno porte que demandam segurança e conectividade entre redes remotas.

Nos testes práticos realizados, o desempenho da solução superou as expectativas teóricas para o *hardware* utilizado. Observou-se uma latência média extremamente baixa (entre 0,2 ms e 0,6 ms) e estabilidade total na conexão (0% de perda de pacotes) durante a transferência de dados criptografados. O monitoramento de recursos evidenciou que o processador do Raspberry Pi operou com folga, indicando que a plataforma possui escalabilidade suficiente para lidar com tráfego mais intenso ou múltiplos clientes simultâneos sem degradação significativa da qualidade de serviço.

4 CAPÍTULO 4

4.1 DESENVOLVIMENTO E IMPLEMENTAÇÃO

Esta seção descreve os procedimentos técnicos adotados para a construção da infraestrutura de VPN *Bridge*. O ambiente foi implementado utilizando dois dispositivos Raspberry Pi 4, configurados respectivamente como Servidor (Sede) e Cliente (Filial Remota), operando sobre o sistema operacional Raspberry Pi OS Lite (32-bit).

4.2 INSTALAÇÃO E DEFINIÇÃO DA ARQUITETURA DO SO

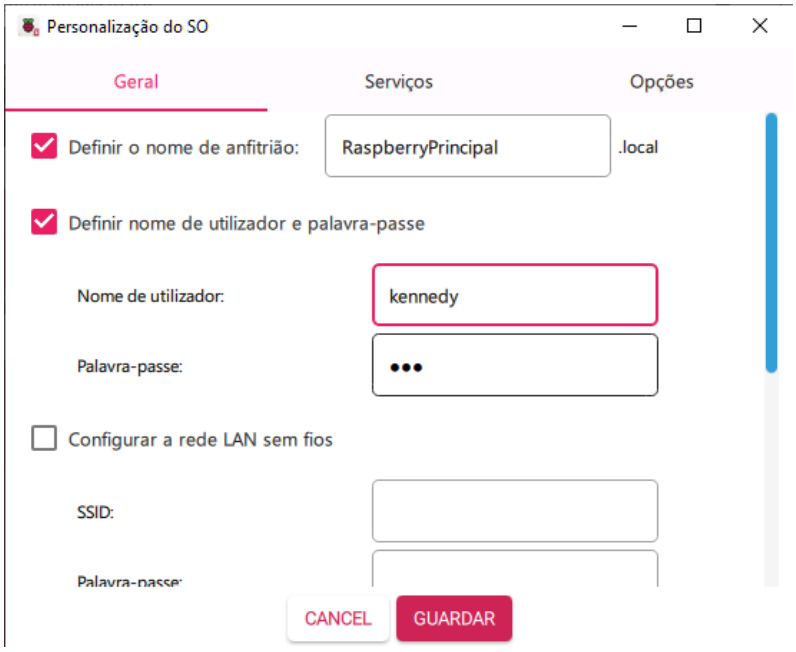
Para a infraestrutura do projeto, optou-se pela utilização do sistema operacional Raspberry Pi OS Lite (32-bit) em ambos os dispositivos. Esta escolha justifica-se pela ausência de interface gráfica de usuário (GUI), o que reduz drasticamente o consumo de memória RAM e processamento, dedicando os recursos do *hardware* exclusivamente para as aplicações de rede.

O processo de gravação da imagem no cartão microSD foi realizado através da ferramenta oficial *Raspberry Pi Imager*. Durante a etapa de gravação, utilizou-se o recurso de "Personalização do SO" (*OS Customization*) para pré-configurar os parâmetros de rede e autenticação.

Conforme demonstrado nas Figuras 1 e 2, a topologia foi definida com funções específicas:

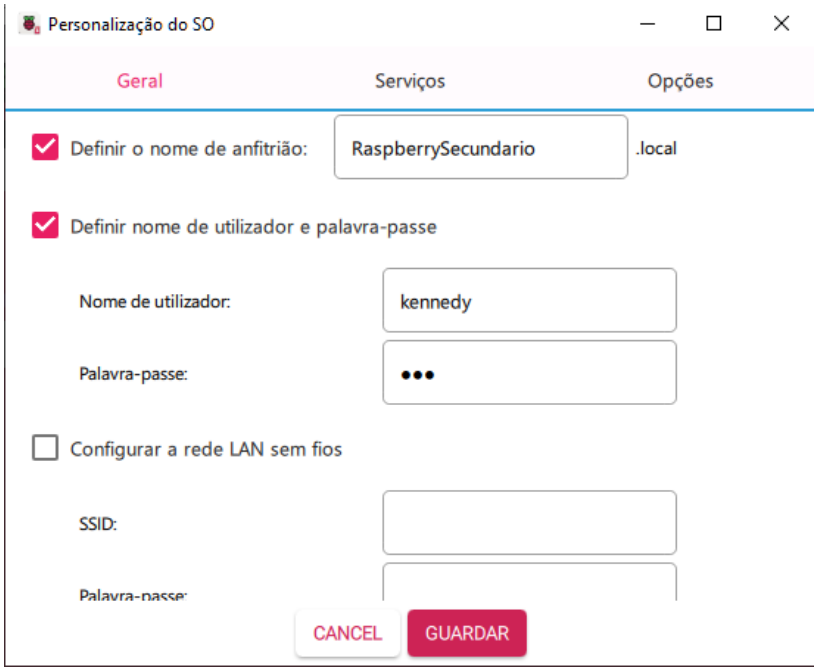
- Servidor: Configurado com o *hostname* RaspberryPrincipal. Este dispositivo será responsável por centralizar os serviços e requisições da rede.
- Cliente**:** Configurado com o *hostname* RaspberrySecundario. Este dispositivo atuará consumindo os recursos ou enviando dados para o servidor.

Figura 1 – Configuração do *hostname* e credenciais do Cliente



Fonte: Foto de arquivo próprio

Figura 2 – Configuração do *hostname* e credenciais do Servidor



Fonte: Foto de arquivo próprio

4.2.1 DEFINIÇÃO DE CREDENCIAIS E SEGURANÇA

Para fins de padronização e facilitação da implementação no ambiente de testes controlado, definiu-se um usuário padrão administrativo idêntico para o Servidor e para o Cliente.

- Usuário (*User*): kennedy
- Senha (*Password*): 123

Ressalta-se que a senha simplificada foi adotada estritamente para a fase de prototipagem acadêmica. Em um ambiente de produção real, recomenda-se a utilização de chaves criptográficas RSA e senhas de alta entropia, conforme as boas práticas de segurança da informação.

4.2.2 ATIVAÇÃO DO PROTOCOLO SSH

O protocolo SSH (*Secure Shell*) é o componente fundamental para a administração deste projeto, pois permite o acesso remoto seguro via linha de comando tanto ao Servidor quanto ao Cliente.

A ativação do serviço foi realizada previamente na aba "Serviços" (*Services*) da ferramenta de personalização do *Raspberry Pi Imager*, habilitando a opção "*Enable* SSH" com autenticação via senha. Isso garante que o *daemon* do SSH (sshd) inicie automaticamente no *boot*, permitindo a gestão imediata dos nós sem a necessidade de periféricos (monitor e teclado).

4.2.3 PROCEDIMENTO DE ACESSO REMOTO E VALIDAÇÃO

Após a inicialização dos dispositivos, o acesso aos terminais é realizado a partir de qualquer computador conectado à mesma sub-rede. O sistema operacional configurado suporta o protocolo mDNS (*Multicast DNS*), permitindo a resolução de nomes locais sem a necessidade de mapeamento prévio de endereços IP.

O comando para acesso via terminal segue a sintaxe *ssh usuario@hostname.local*. Abaixo, detalham-se os comandos para acesso a cada componente da arquitetura:

- Para acessar o Servidor (RaspberryPrincipal):

```
ssh kennedy@RaspberryPrincipal.local
```

- Para acessar o Servidor (RaspberrySecundario):

```
ssh kennedy@RaspberrySecundario.local
```

Ao executar a conexão pela primeira vez, o sistema solicitará a confirmação da impressão digital (*fingerprint*) da chave do *host*. Deve-se confirmar a conexão e inserir a senha configurada (123). Após a autenticação, o terminal remoto estará apto para a instalação dos pacotes de software específicos para as funções de servidor e cliente, respectivamente.

4.3 PREPARAÇÃO DO AMBIENTE E INSTALAÇÃO DE PACOTES

Com o acesso remoto estabelecido, procedeu-se à atualização dos repositórios do sistema e à instalação das ferramentas de rede necessárias em ambos os dispositivos. Optou-se por um conjunto de *softwares* de código aberto amplamente auditados pela comunidade de segurança.

O comando abaixo foi executado tanto no Servidor quanto no Cliente para instalar o OpenVPN (tunelamento), *Easy-RSA* (gestão de chaves), *Bridge-Utils* (manipulação de pontes de rede) e ferramentas de diagnóstico (*tcpdump*, *nmap*, *htop*):

```
sudo apt update && sudo apt install openvpn easy-rsa bridge-utils
iptables-persistent tcpdump nmap htop -y
```

4.4 CONFIGURAÇÃO DO SERVIDOR (RASPBERRYPRINCIPAL)

A configuração do nó central da rede envolveu a criação da infraestrutura de chaves criptográficas e a definição dos parâmetros de tunelamento em Camada 2.

4.4.1 INFRAESTRUTURA DE CHAVES PÚBLICAS (PKI)

Para garantir a autenticidade da conexão, inicializou-se uma PKI própria utilizando o *Easy-RSA*. Foram gerados a Autoridade Certificadora (CA), o certificado do servidor, o certificado do cliente e os parâmetros de Diffie-Hellman. Adicionalmente, gerou-se uma chave estática (ta.key) para autenticação HMAC, visando proteger o servidor contra-ataques de negação de serviço (DoS).

O processo de geração das chaves seguiu uma sequência rigorosa de comandos executados no diretório da Autoridade Certificadora (~/openvpn-ca). Inicialmente, definiram-se as variáveis de ambiente no arquivo vars para garantir a consistência dos metadados dos

certificados. Em seguida, executou-se a sequência de comandos abaixo para a emissão das credenciais digitais:

```
# Inicialização da PKI e construção da CA
./easyrsa init-pki
./easyrsa build-ca nopass

# Geração e assinatura do certificado do Servidor
./easyrsa gen-req server nopass
./easyrsa sign-req server server

# Geração dos parâmetros de Diffie-Hellman e chave HMAC
./easyrsa gen-dh
openvpn --genkey secret ta.key
```

A chave ta.key (HMAC *firewall*) desempenha um papel crítico na segurança: ela permite que o servidor descarte pacotes não assinados antes mesmo de iniciar o processo de *handshake* TLS, mitigando vulnerabilidades de *buffer overflow* e varreduras de porta.

4.4.2 CONFIGURAÇÃO DE REDE E BRIDGE

A arquitetura de VPN *Bridge* (Camada 2) exige que a interface de rede física (eth0) e a interface virtual da VPN (tap0) sejam unificadas em uma interface de ponte lógica. Criou-se a interface br0 no servidor, configurada estaticamente para garantir a persistência do endereço IP 192.168.160.10.

A configuração foi realizada no arquivo /etc/dhcpcd.conf, adicionando-se as seguintes diretivas ao final do arquivo:

```
interface br0
static ip_address=192.168.160.10/24
static routers=192.168.160.254
static domain_name_servers=8.8.8.8 1.1.1.1
```

Esta configuração assegura que, ao reiniciar o dispositivo, a ponte de rede seja recriada automaticamente com o endereçamento correto, fundamental para a disponibilidade do serviço VPN.

4.4.3 PARAMETRIZAÇÃO DO SERVIÇO OPENVPN

O arquivo de configuração principal /etc/openvpn/server/server.conf foi editado para operar em modo TAP sobre o protocolo UDP. Diferente do modo TUN (padrão), a diretiva dev

tap0 permite o encapsulamento de quadros Ethernet completos. O conteúdo final do arquivo configurado foi:

```
# Porta e Protocolo
port 1194
proto udp
dev tap0

# Caminho dos Certificados e Chaves
ca ca.crt
cert server.crt
key server.key
dh dh.pem
tls-auth ta.key 0

# Configuração do Modo Bridge
# Sintaxe: server-bridge [IP-GW] [Máscara] [Início-Pool] [Fim-Pool]
server-bridge    192.168.160.10    255.255.255.0    192.168.160.50
192.168.160.100

# Parâmetros de Criptografia e Segurança
cipher AES-256-CBC
auth SHA512
tls-version-min 1.2

# Scripts de Automação da Ponte
script-security 2
up "/etc/openvpn/up.sh"
down "/etc/openvpn/down.sh"

# Logs e Persistência
keepalive 10 120
persist-key
persist-tun
verb 3
```

Destaca-se a utilização dos *scripts* up.sh e down.sh. Eles são essenciais para automatizar a inserção da interface tap0 na ponte br0 assim que o serviço OpenVPN inicia. O conteúdo do script up.sh criado foi:

```
#!/bin/bash
# Adiciona a interface TAP à Bridge BR0 ao iniciar a VPN
BR=$1
DEV=$2
/sbin/ip link set $DEV up promise on
/sbin/brctl addif $BR $DEV
```

4.5 CONFIGURAÇÃO DO CLIENTE (RASPBERRYSECUNDARIO)

A configuração do cliente focou na capacidade de estabelecer o túnel através de uma rede externa e integrar-se à ponte local de forma transparente. Os certificados (client1.crt, client1.key) gerados no servidor foram transferidos de forma segura via protocolo SCP (*Secure Copy Protocol*) para o cliente.

4.5.1 ARQUIVO DE CONFIGURAÇÃO DO CLIENTE

O arquivo /etc/openvpn/client/client1.conf foi configurado para conectar-se ao endereço IP Público (WAN) da rede do servidor (179.60.128.169), validando o cenário de acesso remoto real simulado através de *Port Forwarding*.

```
client
dev tap
proto udp
# Conexão via IP Público (Simulação de WAN)
remote 179.60.128.169 1194
resolv-retry infinite
nobind

# Certificados
ca ca.crt
cert client1.crt
key client1.key
tls-auth ta.key 1

# Segurança
cipher AES-256-CBC
auth SHA512
key-direction 1

# Script de Automação
script-security 2
up "/etc/openvpn/client/up.sh"
verb 3
```

4.5.2 AUTOMAÇÃO DA INTERFACE DE PONTE NO CLIENTE

Durante os testes iniciais de implementação, observou-se uma falha de sincronização onde a interface tap0 subia, mas não se integrava à ponte br0 a tempo, impedindo o tráfego. Para solucionar este problema técnico, desenvolveu-se um *script* de automação específico (/etc/openvpn/client/up.sh) com uma instrução de espera (*sleep*) para garantir a estabilidade da interface antes da anexação:

```
#!/bin/bash
# Script com delay para garantir a criação da interface
sleep 3
/sbin/brctl addif br0 $1
/sbin/ip link set $1 up
```

Foi necessário conceder permissões de execução ao *script* através do comando sudo chmod +x /etc/openvpn/client/up.sh para que o serviço OpenVPN pudesse executá-lo.

4.6 IMPLEMENTAÇÃO DA SEGURANÇA E FIREWALL

A segurança do perímetro foi estabelecida através da configuração manual do *netfilter* utilizando a ferramenta *iptables*. Optou-se por não utilizar distribuições de *firewall* prontas (como IPFire) para manter o sistema leve e ter controle granular sobre as regras. Adotou-se uma política padrão restritiva (*Default DROP*), bloqueando todo tráfego de entrada e encaminhamento que não fosse explicitamente autorizado. A sequência de comandos executada para aplicar as regras de segurança foi:

```
# 1. Limpar regras antigas
sudo iptables -F
sudo iptables -X

# 2. Definir Política Padrão (Bloquear tudo)
sudo iptables -P INPUT DROP
sudo iptables -P FORWARD DROP
sudo iptables -P OUTPUT ACCEPT

# 3. Permitir tráfego local (Loopback)
sudo iptables -A INPUT -i lo -j ACCEPT

# 4. Permitir conexões já estabelecidas (Stateful Inspection)
sudo iptables -A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -
j ACCEPT

# 5. Liberar porta SSH (Gerenciamento)
sudo iptables -A INPUT -p tcp --dport 22 -j ACCEPT

# 6. Liberar porta OpenVPN (Túnel Criptografado)
sudo iptables -A INPUT -p udp --dport 1194 -j ACCEPT

# 7. Permitir tráfego da Bridge (Essencial para a VPN funcionar)
sudo iptables -A FORWARD -i br0 -o tap0 -j ACCEPT
sudo iptables -A FORWARD -i tap0 -o br0 -j ACCEPT
```

Para garantir que as regras de segurança fossem mantidas após o reinício do sistema, utilizou-se o pacote iptables-persistent, salvando a configuração atual com o comando: sudo netfilter-persistent save

No cliente, enfrentou-se um desafio técnico onde o serviço falhava ao iniciar se a extensão do arquivo fosse. ovpn. O problema foi corrigido renomeando o arquivo de configuração para o padrão .conf exigido pelo *daemon* do Linux: sudo mv client1.ovpn client1.conf

Após a correção, a inicialização automática foi ativada: sudo systemctl enable openvpn-client@client1

Essa configuração final garantiu que, ao ligar os dispositivos, a negociação de chaves, o estabelecimento do túnel, a configuração das pontes de rede e a aplicação das regras de *firewall* ocorressem de forma totalmente autônoma e segura.

5 CAPÍTULO 5

5.1 ANÁLISE DE RESULTADOS

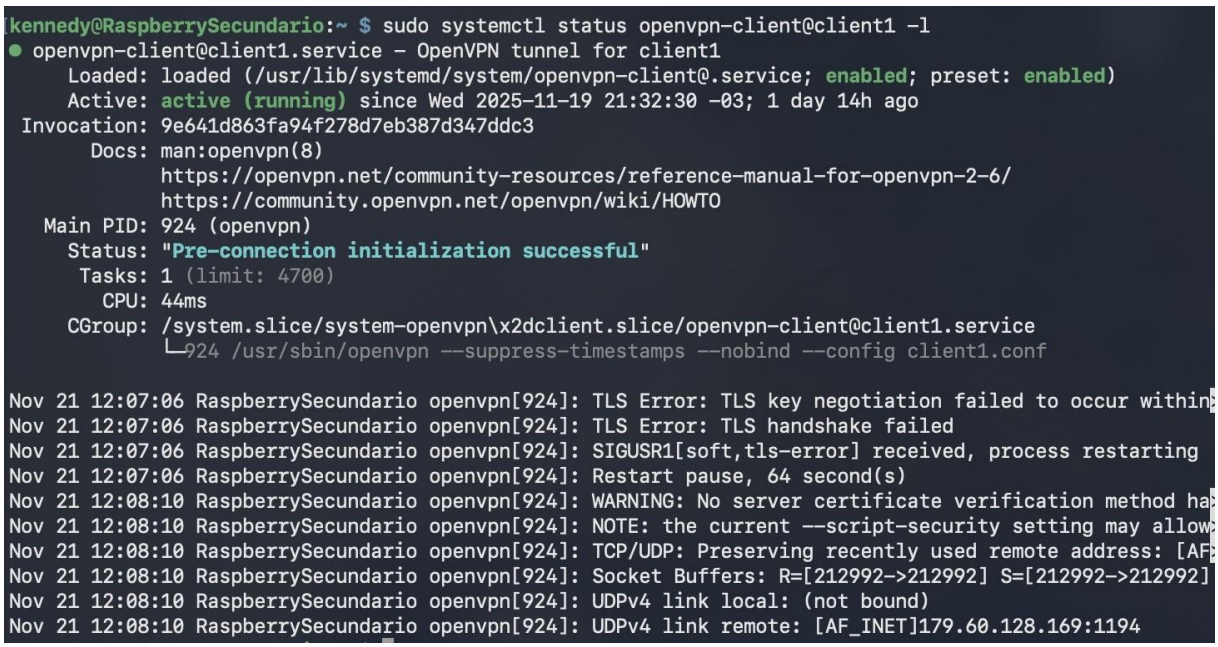
Após a implementação da infraestrutura descrita no capítulo anterior, realizou-se uma bateria de testes para validar a eficácia da solução. Os experimentos focaram em quatro pilares fundamentais definidos nos objetivos específicos: estabelecimento do túnel, transparência de rede (Camada 2), conectividade/latência e desempenho do *hardware*.

5.2 VALIDAÇÃO DO ESTABELECIMENTO DO TÚNEL VPN

O primeiro passo da validação consistiu em verificar se o serviço OpenVPN no cliente foi capaz de estabelecer a conexão com o servidor através do endereço IP público (simulação de WAN).

Utilizou-se o comando systemctl status para auditar o estado do serviço. Conforme demonstrado na Figura 3, o *log* de eventos confirma a sequência de inicialização bem-sucedida ("*Initialization Sequence Completed*"). É importante destacar a linha que indica UDPv4 link remote: [AF_INET]179.60.128.169:1194, comprovando que a conexão foi realizada através do endereço IP externo, validando a configuração de *Port Forwarding* e o acesso remoto via internet.

Figura 3 – Status do serviço OpenVPN indicando conexão via IP Público



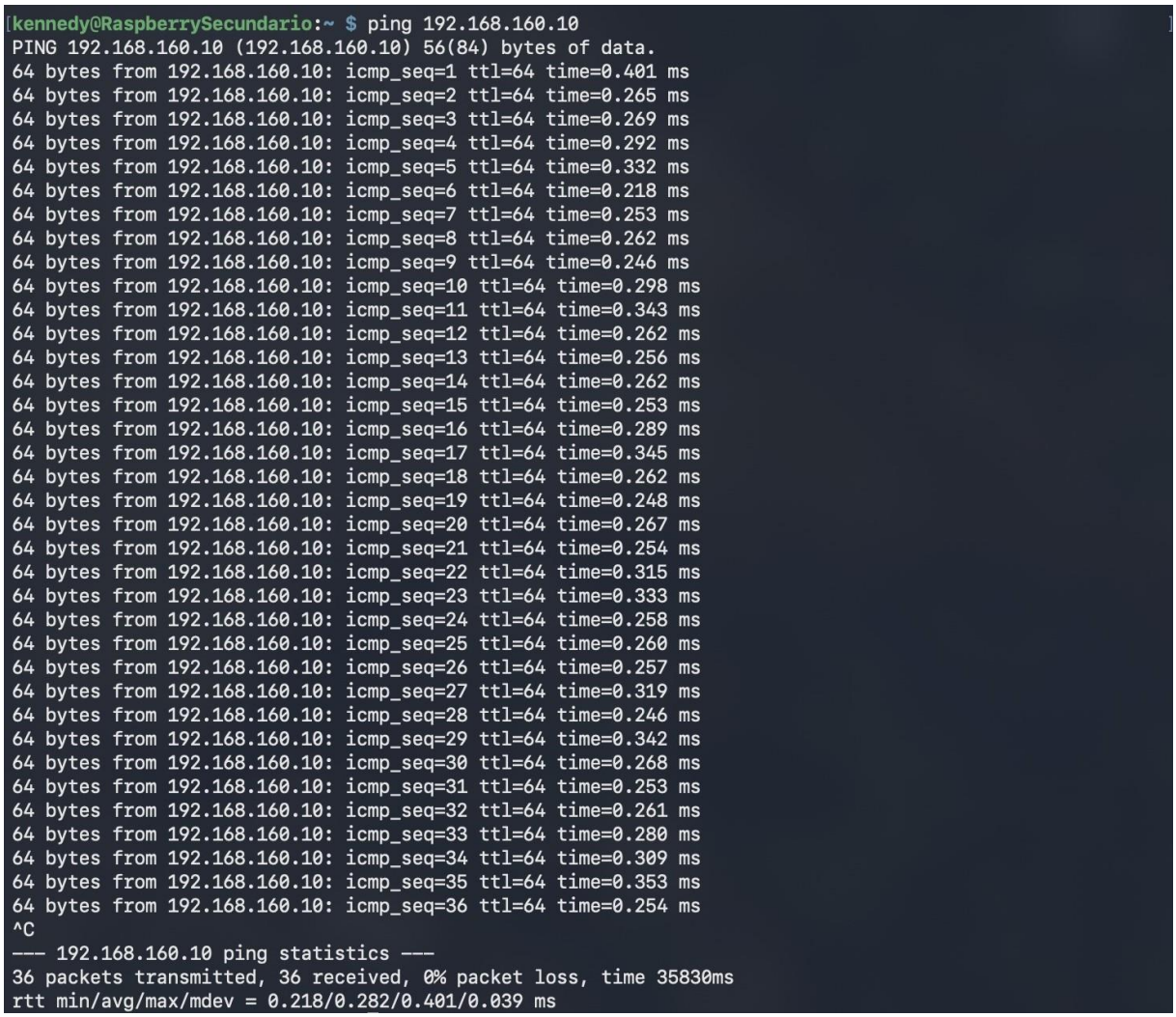
Fonte: Foto de arquivo próprio

5.3 ANÁLISE DE LATÊNCIA E ESTABILIDADE DA CONEXÃO

Para mensurar a qualidade da comunicação através do túnel criptografado, utilizou-se o protocolo ICMP (*Internet Control Message Protocol*). O teste consistiu no envio de 36 pacotes de eco (*ping*) do Cliente (192.168.160.20) para o Servidor (192.168.160.10).

Os resultados, apresentados na Figura 4, indicam uma estabilidade absoluta da conexão, com 0% de perda de pacotes. A latência média (*rtt avg*) registrada foi de 0,282 ms, um valor extremamente baixo. Este resultado evidencia que o *overhead* introduzido pelo encapsulamento do modo TAP e pela criptografia AES-256 não gerou gargalos significativos na transmissão de dados, mantendo a comunicação fluida.

Figura 4 – Teste de conectividade e latência via protocolo ICMP



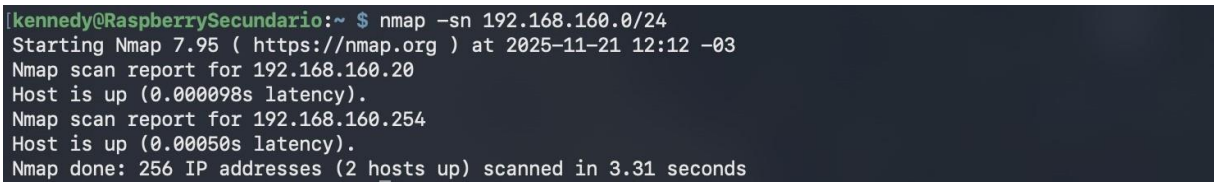
Fonte: Foto de arquivo próprio

5.4 VERIFICAÇÃO DA TRANSPARÊNCIA DE REDE (CAMADA 2)

Um dos diferenciais da VPN *Bridge* é a capacidade de tratar a rede remota como se fosse local. Para validar essa característica, executou-se uma varredura de descoberta de rede utilizando a ferramenta Nmap (*Network Mapper*) a partir do dispositivo cliente.

A Figura 5 comprova a eficácia do modo *Bridge*. O cliente foi capaz de identificar os *hosts* ativos na sub-rede (192.168.160.0/24), incluindo o próprio servidor, através do túnel VPN. Isso confirma que o tráfego de *broadcast* e a descoberta de vizinhança estão operando corretamente, permitindo que dispositivos em redes fisicamente separadas interajam como se estivessem conectados ao mesmo *switch* físico.

Figura 5 – Descoberta de hosts na rede remota comprovando a transparência da Bridge



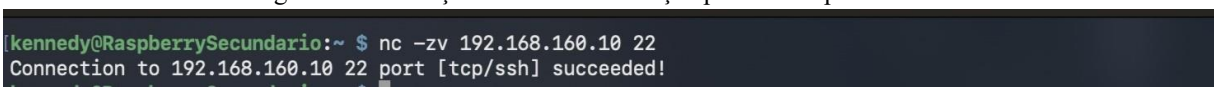
Fonte: Foto de arquivo próprio

5.5 VALIDAÇÃO DAS POLÍTICAS DE SEGURANÇA

Para assegurar que as regras de *firewall* implementadas via *iptables* estavam permitindo o tráfego legítimo enquanto bloqueavam acessos indevidos, realizou-se um teste de conexão TCP na porta 22 (SSH) utilizando a ferramenta Netcat (nc).

A Figura 6 demonstra o resultado "succeeded" (bem-sucedido) ao tentar conectar ao servidor através do túnel VPN. Isso válida a configuração de segurança que permite o tráfego de gerenciamento e de túnel, garantindo que a política padrão restritiva (*DROP*) não impactou a operabilidade dos serviços essenciais.

Figura 6 – Validação de acesso a serviços permitidos pelo Firewall



Fonte: Foto de arquivo próprio

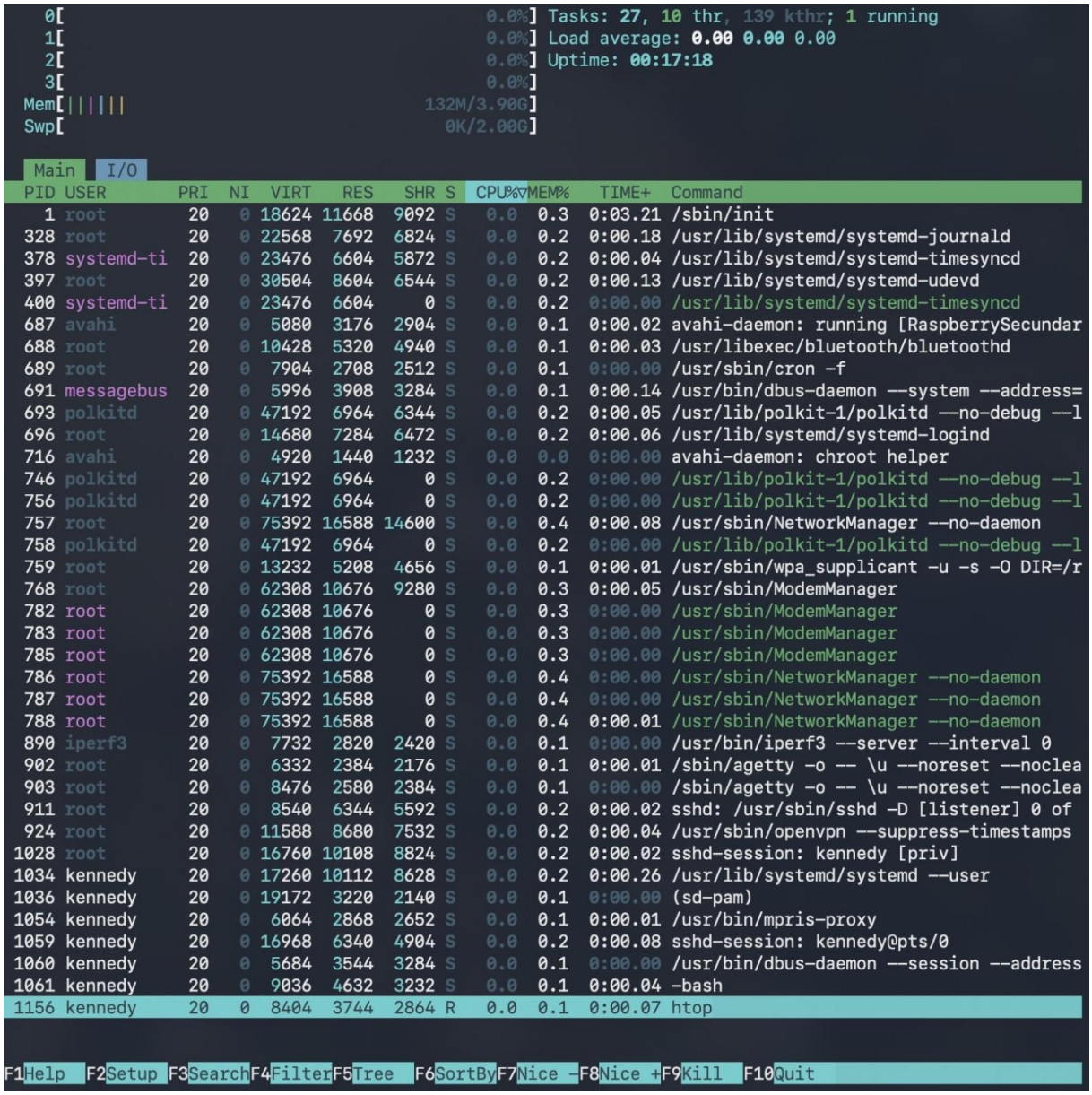
5.6 ANÁLISE DE DESEMPENHO DO HARDWARE

Por fim, analisou-se o impacto computacional da solução no *hardware* do Raspberry Pi 4. Durante a execução dos testes de conectividade e manutenção do túnel criptografado, monitorou-se o consumo de recursos através da ferramenta htop.

Conforme observado na Figura 7, o sistema demonstrou grande eficiência. O consumo de memória RAM permaneceu abaixo de 150MB (em um total de 4GB) e a carga da CPU

manteve-se próxima de zero. Esses dados comprovam que a plataforma escolhida (SBC) possui capacidade de processamento excedente para esta aplicação, validando o Raspberry Pi como uma escolha econômica e tecnicamente viável para atuar como *gateway* de VPN corporativa.

Figura 7 – Monitoramento de recursos do sistema durante a operação



Fonte: Foto de arquivo próprio

5.7 IMPLEMENTAÇÃO DO *DASHBOARD* DE MONITORAMENTO *WEB*

Para complementar a infraestrutura de rede e segurança, desenvolveu-se uma solução de monitoramento centralizado baseada em arquitetura *web*. O objetivo deste módulo foi prover uma interface visual intuitiva para o acompanhamento em tempo real do estado de

saúde dos dispositivos (*Health Check*), consumo de recursos de *hardware* e identificação dos *hosts* conectados à VPN *Bridge*.

5.7.1 ARQUITETURA DA APLICAÇÃO DE MONITORAMENTO

A solução foi construída sobre uma arquitetura desacoplada, utilizando o modelo *Backend-as-a-Service* (BaaS). A estrutura lógica, ilustrada na implementação, divide-se em três camadas:

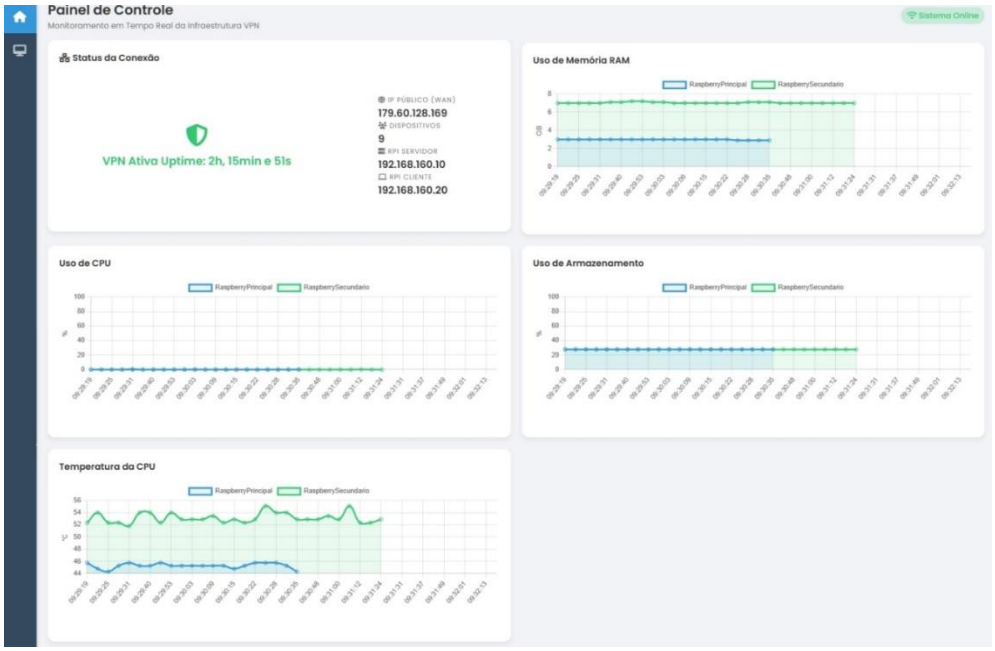
- Agentes de Coleta (Python): *Scripts* desenvolvidos em Python (monitor_hardware.py e monitor_rede.py) executam localmente em cada Raspberry Pi. Estes agentes utilizam as bibliotecas psutil para extração de métricas de *hardware* (CPU, RAM, Temperatura) e scapy/nmap para varredura de rede. Os dados coletados são enviados assincronamente para a nuvem.
- Banco de Dados em Nuvem (Supabase): Utilizou-se o Supabase (baseado em PostgreSQL) como repositório central de dados. Esta escolha eliminou a necessidade de manter um servidor de banco de dados local no Raspberry Pi, economizando recursos de processamento e garantindo a persistência dos logs mesmo em caso de falha física dos dispositivos.
- Frontend de Visualização (Web): Uma interface responsiva desenvolvida em HTML5, CSS3 e JavaScript, que consome os dados diretamente da API do Supabase para renderizar gráficos e tabelas de status.

5.7.2 INTERFACE DE GERENCIAMENTO E VISUALIZAÇÃO DE MÉTRICAS

O painel principal (*Dashboard*) foi projetado para fornecer uma visão holística da infraestrutura. Utilizando a biblioteca gráfica *Chart.js*, implementou-se a visualização histórica do consumo de memória RAM, processamento (CPU) e temperatura dos dispositivos Servidor e Cliente.

A Figura 8 apresenta a tela principal do sistema. É possível observar o indicador de *status* da VPN, que monitora a conectividade ativa, além dos dados de endereçamento IP (Público e Privado) e o tempo de atividade (*uptime*). Os gráficos permitem identificar gargalos de desempenho ou anomalias térmicas que poderiam indicar sobrecarga devido à criptografia da VPN.

Figura 8 – Dashboard web de monitoramento de recursos e status da VPN



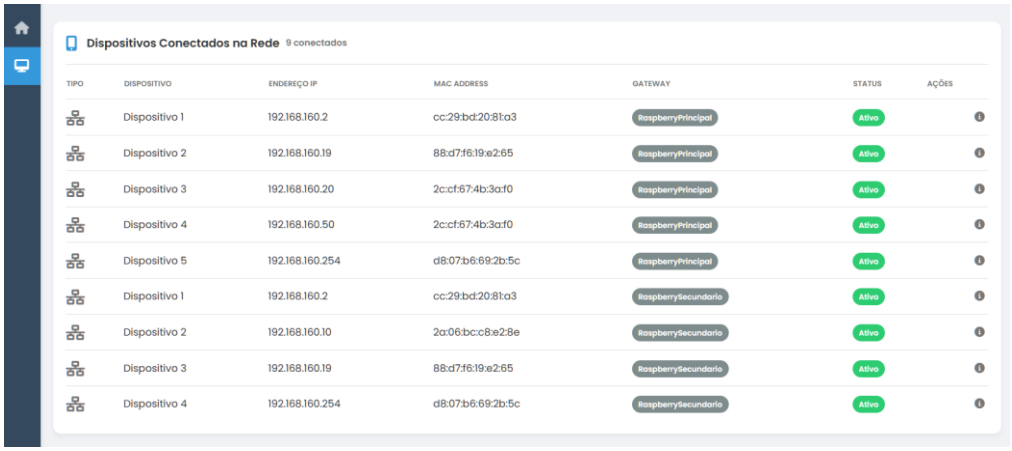
Fonte: Foto de arquivo próprio

5.7.3 MONITORAMENTO DE DISPOSITIVOS CONECTADOS

Para validar a transparência de rede proporcionada pelo modo *Bridge*, desenvolveu-se um módulo específico de listagem de dispositivos. Através da varredura ARP realizada pelos agentes Python na interface de ponte (br0), o sistema identifica todos os equipamentos conectados em ambas as pontas do túnel (rede local e remota).

Conforme demonstrado na Figura 9, a interface lista o Nome, Endereço IP, Endereço MAC e o *Gateway* de origem de cada dispositivo. Esta funcionalidade comprova visualmente a unificação das redes, permitindo que o administrador visualize, em uma única tela, tanto os dispositivos da matriz quanto os da filial conectada via VPN

Figura 9 – Listagem de dispositivos conectados através da VPN *Bridge*



Fonte: Foto de arquivo próprio

6 CONSIDERAÇÕES FINAIS

A execução deste trabalho de conclusão de curso culminou no desenvolvimento, implementação e validação de uma infraestrutura segura para a interconexão de redes disjuntas, utilizando dispositivos de baixo custo e *software* de código aberto. O objetivo principal, que consistia em estabelecer um canal de comunicação estável e transparente entre segmentos de rede distintos, superando as limitações físicas de distanciamento geográfico, foi plenamente alcançado. A solução entregue não apenas resolveu o problema de conectividade proposto, mas o fez garantindo requisitos críticos de segurança, integridade e gerenciabilidade, validando a hipótese de que arquiteturas baseadas em *hardware* acessível podem desempenhar funções de rede tradicionalmente delegadas a equipamentos proprietários de alto custo.

No que tange à análise técnica dos resultados, a escolha arquitetural pelo modo TAP (*Bridge*) no OpenVPN revelou-se o fator determinante para o sucesso da integração em Camada 2. Diferente das abordagens convencionais de roteamento (modo TUN), essa configuração permitiu o tráfego de protocolos de descoberta e *broadcast*, fazendo com que os dispositivos remotos fossem reconhecidos como integrantes da mesma rede local física. Os testes práticos, evidenciados pelas varreduras de rede e pela propagação das tabelas ARP através do túnel, comprovaram essa transparência. Essa característica elimina a complexidade de configurações adicionais de NAT (*Network Address Translation*) entre as pontas, simplificando drasticamente a administração da rede para o usuário final e permitindo a interoperabilidade de sistemas legados que dependem de visibilidade de rede local.

Simultaneamente, a implementação das regras de *firewall* baseadas em *iptables* cumpriu seu papel fundamental de blindagem do perímetro. A política restritiva adotada (*Default DROP*) bloqueou sistematicamente tentativas de conexão não autorizadas, permitindo exclusivamente o tráfego criptografado da VPN e o gerenciamento via SSH. A adoção da criptografia simétrica AES-256, aliada à autenticação mútua via certificados digitais x.509 e à verificação de integridade HMAC (*Hash-based Message Authentication Code*), garantiu que a confidencialidade e a integridade das informações fossem preservadas, mesmo com os dados trafegando por um meio público e hostil como a internet.

Além da segurança e conectividade, a incorporação de um sistema de monitoramento web centralizado elevou a maturidade da solução, conferindo observabilidade em tempo real à infraestrutura. Os indicadores de desempenho coletados através desta interface refutaram a preocupação inicial sobre possíveis gargalos de *hardware* em dispositivos embarcados. O Raspberry Pi 4 demonstrou robustez suficiente para lidar com a sobrecarga de processamento

gerada pela criptografia, mantendo a latência em níveis extremamente baixos e sem perda de pacotes. Isso válida a utilização de *Single Board Computers* (SBCs) para aplicações reais de conectividade segura em pequenas e médias empresas (PMEs), oferecendo uma gestão visual comparável a soluções comerciais.

Sob a ótica do impacto econômico e social, este projeto oferece uma contribuição significativa ao demonstrar que soluções de segurança de nível corporativo podem ser democratizadas. Em um cenário onde equipamentos proprietários de VPN e *Firewall* representam um custo proibitivo para muitas PMEs e *startups*, a arquitetura proposta apresenta-se como uma alternativa viável. A redução de custos não se limita apenas à aquisição do *hardware* (CAPEX), mas estende-se aos custos operacionais (OPEX), uma vez que a solução baseada em Linux isenta a organização de taxas recorrentes de licenciamento. Além disso, a flexibilidade do sistema operacional Raspberry Pi OS permite que o dispositivo assuma múltiplas funções, como servidor de aplicação para o próprio *dashboard* de monitoramento, maximizando o retorno sobre o investimento.

Apesar do êxito na implementação, é importante reconhecer as limitações inerentes ao escopo deste trabalho. A arquitetura atual depende que o lado do servidor possua um endereço IP público acessível ou configuração de *Port Forwarding* no roteador de borda, o que pode ser um obstáculo em cenários onde o provedor de internet utiliza CGNAT (*Carrier-Grade NAT*), exigindo o uso de intermediários externos. Adicionalmente, embora o Raspberry Pi 4 tenha suportado com folga o tráfego de um cliente, o desempenho em cenários com dezenas de conexões simultâneas e tráfego intenso na ordem de Gigabits seria limitado pela capacidade física da interface de rede e pelo barramento da CPU, não substituindo *appliances* dedicados em grandes *data centers*. Outro ponto de atenção é a gestão manual da PKI via linha de comando, que pode se tornar complexa à medida que a rede escala.

Visando a evolução contínua desta infraestrutura, sugerem-se implementações futuras que ampliem a robustez do sistema. Recomenda-se o desenvolvimento de um mecanismo de alta disponibilidade com redundância de *link* (*Failover*), utilizando múltiplas conexões de internet para garantir a continuidade do serviço em caso de falha de um provedor. A integração de um Sistema de Detecção e Prevenção de Intrusões (IDS/IPS), como Snort ou Suricata, diretamente no *gateway*, adicionaria uma camada de inteligência capaz de analisar o tráfego dentro do túnel. Também seria benéfica a expansão do *dashboard* web para incluir funcionalidades de administração, como a revogação de certificados e o bloqueio de dispositivos, além do suporte nativo ao protocolo IPv6.

Conclui-se, portanto, que a solução desenvolvida oferece uma alternativa eficiente, econômica e tecnicamente sólida para a integração de filiais, escritórios domésticos ou ambientes de IoT industrial. O sistema entregue não apenas resolve o problema de conectividade entre redes separadas, mas o faz com um nível de segurança, transparência de rede e capacidade de monitoramento que geralmente só é encontrado em equipamentos proprietários de custo elevado. A validação prática deste Trabalho de Conclusão de Curso reforça o potencial do *Open Source* e do *Open Hardware* como pilares fundamentais para a inovação, acessibilidade e segurança da informação na engenharia moderna.

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

7 REFERÊNCIAS:

ANDRESS, Jason. ***The Basics of Information Security****: Understanding the Fundamentals of InfoSec in Theory and Practice. 2. ed. Elsevier, 2014.*

CHESWICK, Bill; BELLAVISTA, Steven; RUBEN, Aviel. ***Firewalls and Internet Security****: Repelling the Wily Hacker. 3. ed. Boston: Addison-Wesley, 2021.*

CISCO. ***VPN Bridge for Branch Connectivity****: Case Study. Cisco Systems, 2022.*

DEBIAN PROJECT. ***Sobre o Debian****. Disponível em: https://www.debian.org/intro/about.pt.html. Acesso em: 08 abr. 2025.*

GIL, Antonio Carlos. ***Como elaborar projetos de pesquisa****. 6. ed. São Paulo: Atlas, 2017.*

IEEE. ***Performance of Layer 2 VPNs on SBCs****: A Case Study with Raspberry Pi. IEEE Communications Surveys & Tutorials, 2023.*

LUNDUKE, Bryan. ***Self-hosted Server Applications with Raspberry Pi and Linux****. Apress, 2022.*

NASCIMENTO, Davi. ***Fundamentos de Segurança da Informação****. São Paulo: Novatec, 2016.*

OPENVPN. ***Bridging and Routing****. OpenVPN Documentation. Disponível em: https://openvpn.net/community-resources/ethernet-bridging/. Acesso em: 08 abr. 2025.*

RASPBERRY PI FOUNDATION. ***Raspberry Pi 4 Model B****. Disponível em: https://www.raspberrypi.com/products/raspberry-pi-4-model-b/. Acesso em: 10 abr. 2025.*

RASPBERRY PI FOUNDATION. ***Raspberry Pi Hardware Documentation****. Disponível em: https://www.raspberrypi.com/documentation/computers/processors.html. Acesso em: 25 nov. 2025.*

SOUZA, Ricardo. ***Plano de Continuidade de Negócios em Ambientes Virtuais****. São Paulo: Atlas, 2020.*

STALLINGS, William. ***Segurança em redes****: princípios e práticas. 5. ed. São Paulo: Pearson, 2017.*

SWAN, Travis. ***Practical VPNs****: Building and Configuring Secure Virtual Private Networks. O'Reilly Media, 2014.*

ANEXO 1 – Termo de autorização de publicação de produção acadêmica.

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS
GABINETE DO REITOR
Av. Universitária, 1069 • Setor Universitário
Caixa Postal 86 • CEP 74605-010
Goiânia • Goiás • Brasil
Fone: (62) 3946.1000
www.pucgoias.edu.br • reitoria@pucgoias.edu.br

RESOLUÇÃO nº 038/2020 – CEPE

ANEXO I

APÊNDICE ao TCC

Termo de autorização de publicação de produção acadêmica

O(A) estudante Kennedy Fernandes dos Santos
do Curso de Engenharia de computação, matrícula 2019.1.0033.0078-0,
telefone: 62 9 9826-0249 e-mail Kennedy.fernandes.dos.santos@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 Integração Segura de Redes distintas com FireWall e Raspberry Pi usando VPN Brig 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, 30 de setembro de 2025.

Assinatura do autor: Kennedy F. dos Santos
Nome completo do autor: Kennedy Fernandes dos Santos

Assinatura do professor–orientador: _____
Nome completo do professor-orientador: Marcelo Antonio Cabral de Araújo