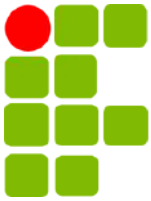


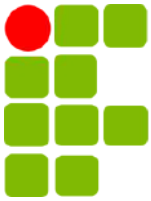
Instituto Federal de Educação, Ciência e Tecnologia da Paraíba
Campus Campina Grande
Coordenação do Curso Superior de Tecnologia em Telemática

Um Estudo de Migração de *Firewall* no IFPB, *campus* Campina Grande.

WESLEY PORTO SANTOS

Orientador: Marcelo Portela Sousa
Coorientador: Átila de Souza Medeiros

Campina Grande, fevereiro de 2025



Instituto Federal de Educação, Ciência e Tecnologia da Paraíba
Campus Campina Grande
Coordenação do Curso Superior de Tecnologia em Telemática

Um Estudo de Migração de *Firewall* no IFPB, *campus* Campina Grande.

WESLEY PORTO SANTOS

Monografia apresentada à Coordenação do Curso de Telemática do IFPB - *Campus* Campina Grande, como requisito parcial para conclusão do curso de Tecnologia em Telemática.

Orientador: Dr. Marcelo Portela Sousa
Coorientador: Me. Átila de Souza Medeiros

Campina Grande, fevereiro de 2025

Catalogação na fonte:
Ficha catalográfica elaborada por Gustavo César Nogueira da Costa - CRB 15/479

S237e Santos, Wesley Porto
Um estudo de migração de firewall no IFPB, campus Campina Grande / Wesley Porto Santos. - Campina Grande, 2025.
80 f. : il.

Trabalho de Conclusão de Curso (Curso Superior de Tecnologia em Telemática) - Instituto Federal da Paraíba, 2024.
Orientador: Dr. Marcelo Portela Sousa.
Coorientador: Me. Átila de Souza Medeiros.

1. Telemática 2. Rede de computadores - Segurança de redes 3. Firewall 4. Administração de redes 5. IFPB Campus Campina Grande - segurança e administração de redes I. Sousa, Marcelo Portela II Medeiros, Átila de Souza III.Título.

CDU 004.7

Um Estudo de Migração de *Firewall* no IFPB, *campus* Campina Grande.

WESLEY PORTO SANTOS

Dr. Marcelo Portela Sousa
Orientador

Me. Átila de Souza Medeiros
Coorientador

Dr. David Candeia Medeiros Maia
Membro da Banca

Dr. Petrônio Carlos Bezerra
Membro da Banca

Campina Grande, Paraíba, Brasil
Fevereiro/2025

Esse trabalho é dedicado a todos aqueles que me incentivaram, ajudaram e fizeram que com a trajetória até aqui fosse possível.

Os ideais que iluminaram o meu caminho são a bondade, a beleza e a verdade.
Albert Einstein

Agradecimentos

Chegar até aqui não seria possível sem ajuda, dito isso, dedico meus mais profundos agradecimentos, primeiramente, a minha família, tios, primos, que me apoiaram, me incentivaram, foram pacientes e compreensivos em todos os momentos. Minha amada esposa Mércia, que me deu todo o suporte, especialmente emocional, com seu infinito cuidado e amor, juntamente com meu amado filho Guilherme.

A minha amada mãe, Mônica, que sempre me ensinou a ser uma pessoa melhor, dia após dia, com seu amor e cuidado incondicional. Meu amado pai, meu maior herói Ivanildo (*in memoriam*), que não está mais nesse plano, nem está presente em corpo, mas sim em espírito. É muito difícil esperar, mas vou cobrar o abraço aí do outro lado. Minha amada irmã, Ingrid, a outra metade de mim. Sem vocês, seria impossível sequer ter começado. Obrigado por me tornar a maior parte do que sou hoje.

A todos os meus amigos, pela cobrança intensa pra chegar aqui, por toda a psicologia reversa envolvida e aplicada para que eu me formasse, obrigado serviu mesmo. Agradeço em especial a meu amigo Erick Pereira de Albuquerque e ao professor Jerônimo da Silva Rocha, que me fizeram ver que o curso que eu precisava fazer era o de Tecnologia em Telemática. A José Silvestre da Silva Galvão (ou como digo, Galvões), amigo e companheiro de equipe na Huawei ICT *Competition* 2023-2024, por toda a sua disponibilidade e ajuda antes e durante esse trabalho. Dannylo Xavier Wanderley, amigo, companheiro de trabalho e jogos eletrônicos, pela prontidão e verdadeira vontade em ajudar. A meu amigo, Eduardo Jerônimo de Moura Júnior pela ajuda e incentivos desde o começo. A meu grande amigo Alexandre Pires de Sousa, pela parceria de irmão e por todo apoio logístico do dia a dia, sempre prestativo e disponível.

Quero expressar minha imensa gratidão ao amigo, professor e orientador, Dr. Marcelo Portela Sousa, ao amigo e coorientador Me. Átila de Souza Medeiros, por suas excepcionais orientações e mentorias. O suporte e suas valiosas contribuições, para o desenvolvimento desse trabalho foram fundamentais. Sem vocês na orientação, ainda estaria perdido nesse imenso mar acadêmico.

Agradeço imensamente também, a todos os professores que tive a honra de ser aluno durante esse curso, com os conhecimentos passados por vocês, hoje sou mais capaz do que já fui antes e tenham em mim, um amigo. Agradeço também a essa casa, chamada Instituto Federal de Educação, Ciência e Tecnologia da Paraíba, meu IFPB. Da portaria até a reitoria, obrigado pelo acolhimento, suporte, cuidado e carinho.

Por fim, agradeço aos meus colegas e amigos que fiz durante esses anos e que também me ajudaram nessa jornada acadêmica, tornado essa experiencia muito mais interessante e prazerosa.

Resumo

Instalada em 2009, a infraestrutura de *firewall* do IFPB, *campus* Campina Grande, vem enfrentando problemas causados por instabilidades e exposição a vulnerabilidades devido a falta de atualizações de *hardware* e *software* ao longo dos anos. Além disso, foram identificadas limitações nesse sistema legado, incluindo a falta de funcionalidades avançadas de monitoramento, análise de tráfego aprimorado e recursos de gerência. Com isso, este trabalho realiza o planejamento da migração de um *firewall* legado (**iptables**) para uma solução de *firewall* de próxima geração (NGFW), com o objetivo de aprimorar a segurança da rede. O estudo aborda a importância da modernização da infraestrutura de segurança em resposta às novas ameaças cibernéticas e à crescente complexidade do ambiente digital. O estudo também inclui a avaliação das necessidades específicas da organização, o planejamento da nova arquitetura de *firewall* e a simulação do *design* proposto no simulador PNETLab da solução NGFW OPNsense operando em alta disponibilidade (*high availability* –HA) para validar a proposta. Os resultados indicam que o *design* proposto com o OPNsense operando em HA atende às demandas institucionais, proporciona maior robustez, simplifica a administração e aprimora a segurança da rede, contribuindo para a continuidade das operações e a proteção de dados sensíveis.

Palavras-chave: *Firewall*, Alta disponibilidade, OPNsense, iptables.

Abstract

Installed in 2009, the firewall infrastructure of IFPB, Campina Grande campus, has been facing problems caused by instability and exposure to vulnerabilities due to the lack of hardware and software updates over the years. In addition, limitations were identified in this legacy system, including the lack of advanced monitoring functionalities, improved traffic analysis and management resources. Therefore, this work plans the migration of a legacy firewall (iptables) to a next-generation firewall (NGFW) solution, with the objective of improving network security. The study addresses the importance of modernizing the security infrastructure in response to new cyber threats and the increasing complexity of the digital environment. The study also includes the assessment of the specific needs of the organization, the planning of the new firewall architecture and the simulation of the proposed design in the PNETLab simulator of the OPNsense NGFW solution operating in high availability (HA) to validate the proposal. The results indicate that the proposed design with OPNsense operating in HA meets institutional demands, provides greater robustness, simplifies administration and improves network security, contributing to the continuity of operations and the protection of sensitive data.
Keywords: *Firewall*, *High Availability – HA*, OPNsense, iptables.

Sumário

Lista de Abreviaturas **xiii**

Lista de Figuras **xiv**

Lista de Tabelas **xvi**

1 Introdução **1**

1.1 Motivação e definição do problema 2

1.2 Objetivos 3

1.2.1 Objetivo geral 3

1.2.2 Objetivos específicos 3

2 Fundamentação teórica **4**

2.1 Trabalhos relacionados 4

2.2 *Firewall* 6

2.2.1 *Packet filtering firewall* 6

2.2.2 *Stateful inspection firewalls* 7

2.2.3 *Application-level firewalls* 8

2.2.4 *Next-generation firewalls* 8

2.3 OPNsense 9

2.4 O iptables 9

2.5 Alta disponibilidade (*High Availability* - HA) 10

2.6 PNETLab 12

2.7 LAGG – (*Link Aggregation*) 12

2.7.1 LACP – (*Link Aggregation Control Protocol*) 13

2.8 VLAN – (*Virtual Local Area Network*) 13

2.9 *Virtual Machine* – VM 14

2.10 Protocolo ICMP 15

3 Metodologia **16**

3.1 Topologia e arquitetura do cenário atual 21

3.2 Topologia e arquitetura proposta 22

3.3 Instalação OPNsense 23

3.4 Configuração do ambiente de HA 29
3.4.1 Requisitos 29
3.4.2 Configuração das interfaces de rede 29
3.4.3 Configuração do CARP 31
3.4.4 Configuração do NAT 36
3.4.5 Ativação do HA 43
3.4.6 Testes do ambiente de HA 46
3.5 Migração das regras e políticas de *firewall* 47

4 Resultados dos experimetos **48**
4.1 Resultados com a topologia atual do *campus* 48
4.2 Resultados com a topologia proposta no trabalho 52

5 Considerações finais e sugestões para trabalhos futuros **59**
5.1 Sugestões para trabalhos futuros 60

A Questionário **61**

Referências Bibliográficas **65**

Lista de Abreviaturas

CARP	*Common Address Redundancy Protocol*
HA	*High Availability* (Alta Disponibilidade)
PNETLab	*Packet Network Emulator Tool Lab*
LAGG	*Link Aggregation*
LACP	*Link Aggregation Control Protocol*
WAN	*Wide Area Network*
LAN	*Local Area Network*
VLAN	*Virtual Local Area Network*
ISO	*International Organization for Standardization*
OSI	*Open Systems Interconnection*
VM	*Virtual Machine*
MB	*Megabytes*
GB	*Gigabytes*
GUI	*Graffical User Interface*
CLI	*Command Line Interface*
VIP	*Virtual IP*
VHID	*Virtual Host Identificator*

Lista de Figuras

2.1 Exemplo da arquitetura utilizada no CARP 12

3.1 Recursos da VM de firewall atual 16
3.2 Topologia e arquitetura atual 21
3.3 Topologia e arquitetura da solução proposta. 22
3.4 Instalação OPNsense - Passo 1. 24
3.5 Instalação OPNsense - Passo 2. 24
3.6 Instalação OPNsense - Passo 3. 25
3.7 Instalação OPNsense - Passo 4. 25
3.8 Instalação OPNsense - Passo 5. 26
3.9 Instalação OPNsense - Passo 6. 26
3.10 Instalação OPNsense - Passo 7. 27
3.11 Instalação OPNsense - Passo 8. 27
3.12 Instalação OPNsense - Passo 9. 28
3.13 Instalação OPNsense - Passo 10. 28
3.14 Endereços IP OPN-FW-01. 30
3.15 Endereços IP OPN-FW-02. 31
3.16 Configuração dos VIPs 32
3.17 Adição dos VIPs 33
3.18 Configuração do VIP da interface WAN 33
3.19 VIP da interface WAN configurada 34
3.20 Configuração dos VIP para uma das interfaces LAN 34
3.21 Configuração dos VIP no OPN-FW-01 35
3.22 Configuração dos VIP no OPN-FW-02 35
3.23 Acesso ao menu de configuração do NAT Outbound 36
3.24 Alteração do modo de NAT Outbound 36
3.25 Criação de alias para redes de origem. 37
3.26 Adição dos aliases. 38
3.27 Definição das informações do alias. 38
3.28 Visualização do alias criado. 39
3.29 Adição da regra de NAT. 40

3.30 Configuração da regra de NAT através da interface de HA. 41
3.31 Configuração da regra de NAT através da interface WAN. 42
3.32 Visualização das regras de NAT criadas. 43
3.33 Ativação o HA no firewall master. 43
3.34 Configuração dos parâmetros do HA no firewall master. 44
3.35 Ativação do HA no firewall backup. 45
3.36 Verificação do status do HA no firewall master. 46
3.37 Verficação do status do HA no firewall backup. 46

4.1 Teste de ping para os destinos citados. 49
4.2 Gráfico com o resultado do teste para os destinos citados. 49
4.3 Firewall desligado por falha. 50
4.4 Teste de ping para os destinos citados. 51
4.5 Gráfico com o resultado do teste para os destinos citados. 51
4.6 Teste de ping para os destinos citados anteriormente. 52
4.7 Gráfico com o resultado do teste para os destinos citados anteriormente. . . . 53
4.8 Momento da interrupção do firewall master. 54
4.9 Perda de pacotes detectada. 55
4.10 Perda de pacotes detectada. 56
4.11 Pacotes gerados com intervalo 1s. 56
4.12 Pacotes gerados com intervalo 100ms. 57
4.13 Pacotes gerados com intervalo 10ms. 57
4.14 Firewall backup assumindo o papel de master no CARP. 58

Lista de Tabelas

3.1 Comparação entre OPNsense, pfSense e iptables 18
3.2 Pré-requisitos de Hardware recomendados. 23
3.3 Tabela de Endereçamento dos Firewalls. 29
3.4 Configuração dos endereços VIP. 31
3.5 Chains iptables. 47
3.6 NAT iptables. 47

Capítulo 1

Introdução

As inovações tecnológicas transformaram a maneira como as pessoas vivem, trabalham, se divertem, compartilham informações e se comunicam. Com a crescente adoção da *Internet* como uma ferramenta essencial para negócios — seja para vendas, colaboração ou comunicação — as aplicações *web* tornaram-se um dos elos mais frágeis na estratégia de cibersegurança das organizações [Sheth e Thakker 2011].

Os *firewalls* de rede desempenham um papel fundamental na proteção de redes confiáveis contra redes não confiáveis, filtrando o tráfego com base em políticas de segurança predefinidas. Geralmente, um *firewall* é instalado na entrada de cada rede privada conectada à *Internet*. Sua principal função é inspecionar cada pacote de dados que passa por essa entrada e decidir se ele deve ser aceito e encaminhado ou bloqueado e descartado.

A função essencial de um *firewall* é controlar o fluxo de tráfego entre redes com diferentes níveis de confiança. Um exemplo típico é a separação entre a *Internet*, considerada uma zona sem confiança, e uma rede interna, que possui um nível mais alto de segurança. Já uma zona intermediária, situada entre a *Internet* e a rede interna confiável, é comumente chamada de "rede de perímetro" ou zona desmilitarizada (*Demilitarized Zone* — DMZ) [Lutkevich 2021]

Além dos fatores de segurança em redes, há também a necessidade de garantir disponibilidade contínua e evitar interrupções nos sistemas computacionais. Isso tem tornado a implementação de soluções com alta disponibilidade (*high availability* — HA) mais relevante. Esse objetivo pode ser alcançado por meio da adoção de recursos redundantes de *software* e *hardware*, eliminando pontos únicos de falha e aumentando a resiliência da infraestrutura de tecnologia da informação (TI) [Godoy 2016].

Neste sentido, o presente trabalho de conclusão de curso (TCC) realiza o planejamento da migração de um *firewall* legado (**iptables**) para uma solução de *firewall* de próxima geração (*Next-Generation Firewall* – NGFW) no âmbito do IFPB, *campus* Campina Grande.

Com o objetivo de aprimorar a segurança da rede, propõe-se uma arquitetura de alta disponibilidade utilizando uma solução NGFW de código aberto baseado no FreeBSD e desenvolvido pela empresa holandesa Decisio, denominada OPNSense [OPNSense 2024].

Para validar a proposta, utilizou-se a ferramenta de simulação PNETLab, virtualizada em ambiente MS Windows.

Os resultados iniciais indicam que o *design* proposto com o OPNSense operando em HA, atende as demandas institucionais, proporciona maior robustez, simplifica a administração e aprimora a segurança da rede, contribuindo para a continuidade das operações e a proteção de dados sensíveis, como também, garante a escalabilidade da solução.

1.1 Motivação e definição do problema

Como os imperadores chineses, que construíram a Grande Muralha da China, para proteger seu território contra as ameaças externas [Kiratsata *et al.* 2022], analogamente, um sistema de *firewall* deve proteger os perímetros internos contra as ameaças provenientes da *Internet*. Ataques e violações de segurança cibernética são eventos bastante comuns, ressaltando a necessidade de estabelecer uma linha de defesa para reforçar a segurança e a resiliência do ciberespaço contra ameaças cada vez mais sofisticadas [Monrose e Werner 2018].

Preocupados com isso, entre os anos de 2013 e 2014, a equipe de TI do *campus*, implementou um sistema de *firewall*, com o objetivo de garantir a segurança da rede, controlar acessos e proteger os serviços internos contra ameaças. Durante anos, essa solução desempenhou um papel fundamental na estrutura de segurança da organização, permitindo a filtragem de tráfego e a prevenção de ataques cibernéticos.

No entanto, com o avançar do tempo e o crescimento da demanda por conectividade, esse *firewall* começou a apresentar falhas recorrentes, comprometendo a estabilidade da rede. Tais falhas resultam em instabilidades, lentidão no tráfego de dados e, em alguns casos, interrupções completas do acesso à *Internet* e dos serviços internos, impactando diretamente as atividades administrativas e acadêmicas do *campus*.

Além disso, a obsolescência do *hardware* e a limitação dos recursos de *software* do sistema atual (**iptables**), impedem a implementação de novas técnicas de segurança e a adaptação às ameaças cibernéticas mais modernas. A ausência de suporte adequado e a dificuldade de atualização também são fatores que agravam o problema, tornando a substituição dessa solução de *firewall*, uma necessidade urgente.

Diante desse cenário, torna-se essencial a implementação de uma nova solução de *firewall* que atenda às exigências atuais de segurança e desempenho. A proposta para a substituição visa adotar um sistema mais robusto, com maior capacidade de processamento, tolerância a falhas, funcionalidades avançadas de detecção de ameaças e gestão eficiente do tráfego de rede.

1.2 Objetivos

1.2.1 Objetivo geral

Planejar a migração dos serviços legados de *firewall* em execução na infraestrutura de TI do IFPB, *campus* Campina Grande, para uma plataforma de *firewall* baseada em FreeBSD/Unix, com suporte a tecnologias de alta disponibilidade.

1.2.2 Objetivos específicos

- Avaliar o cenário atual de infraestrutura de TI do *campus*;
- Levantar quais serviços legados que precisam ser atualizados para uma nova infraestrutura de TI;
- Pesquisar as principais plataformas de *firewall* baseada em FreeBSD/Unix com suporte a alta disponibilidade disponíveis no mercado;
- Planejar o *design* da migração dos serviços;
- Simular a arquitetura planejada para validar a proposta.

Capítulo 2

Fundamentação teórica

São abordados nesse capítulo, trabalhos relacionados, conceitos fundamentais sobre *firewall*, as principais características sobre o OPNsense, recursos de alta disponibilidade, ambiente de virtualização PNETLab e protocolos de rede que serão utilizados nesse trabalho. O entendimento desses conceitos, é fundamental para o funcionamento e sucesso da implementação proposta nesse trabalho.

2.1 Trabalhos relacionados

A migração de um *firewall* não é apenas uma substituição de *software* ou *hardware*, mas um processo estratégico que envolve análise de riscos, adaptação de políticas de segurança e validação rigorosa. Com isso, a busca por trabalhos similares pode ajudar na tarefa de migração, para vencer eventuais dificuldades encontradas durante o decorrer do processo.

O trabalho de [Godoy 2016] utiliza a abordagem de alta disponibilidade, porém, com a solução de *firewall* do pfSense. Apesar de usar uma solução de *firewall* diferente, o autor precisou fazer os ajustes necessários de arquitetura e topologia anterior, para proporcionar os resultados buscados nesse trabalho.

É apresentado no trabalho de [JÚNIOR e JÚNIOR 2019], um relatório de migração de uma solução *open source* virutal com pfSense, para um *appliance*[1] de *firewall* físico usando Fortigate. Essa solução possui custos para sua aquisição, o que foge do escopo e perfil desse trabalho, e, apesar de ser uma solução mais moderna, não possui suporte a alta disponibilidade da forma que foi instalada, o que é pré-requisito para esse trabalho.

É abordada no trabalho de [Lara 2010], a aplicação de um sistema de *firewall* de camada de aplicação do modelo OSI (camada 7), com objetivo de analisar o tráfego dos protocolos HTTP[2] (*Hypertext Transfer Protocol* – Protocolo de Transferência de Hipertexto)

[1]Um *appliance* é um equipamento dedicado que combina *hardware* e *software* para desempenhar uma função específica dentro de um ambiente de TI (Tecnologia da Informação), geralmente com foco em desempenho, segurança e facilidade de administração.

[2]É um protocolo de comunicação que permite a transferência de hypertexto que usado geralmente em páginas web.

e HTTPS[3] (*Hypertext Transfer Protocol Secure* – Protocolo de Transferência de Hipertexto Seguro). O artigo apresenta o *design* e a aplicação do sistema de *firewall open source*, ModSecurity, para ambientes de alta disponibilidade.

De forma análoga, este trabalho também busca melhorar as ferramentas de gestão e controle de segurança da informação da rede de dados do *campus* Campina Grande do IFPB, diferenciando-se das soluções apresentadas nesta seção, por propor o uso do NGFW OPNSense operando em HA, o que permite a continuidade dos serviços mesmo havendo falhas de *hardware* ou *software* em um dos serviços. Usar o OPNsense como uma solução de NGFW, pode oferecer várias vantagens em comparação a outras opções no mercado (por exemplo: pfSense, ipFire, ClearOS, Untangle, VyOS). Entre elas, elencam-se:

1. **Código Aberto**: OPNsense é uma solução de código aberto, o que significa que há acesso total ao seu código-fonte. Isso permite maior transparência em termos de segurança e funcionalidade. Além disso, é possível modificar o software conforme suas necessidades;

2. **Custo**: Sendo gratuito e de código aberto, o OPNsense pode ser uma opção mais econômica, especialmente para pequenas e médias empresas que buscam soluções robustas sem altos custos de licenciamento;

3. **Interface Intuitiva**: OPNsense oferece uma interface de usuário amigável e bem projetada, facilitando a configuração e o gerenciamento, mesmo para aqueles que não são especialistas em redes;

4. **Recursos Avançados**: Inclui muitos recursos de segurança e gerenciamento, como IDS[4]/IPS[5], VPN[6], filtragem de conteúdo, balanceamento de carga e suporte a múltiplas *WANs*, que competem com as soluções proprietárias;

5. **Atualizações Regulares**: OPNsense recebe atualizações regulares que não apenas corrigem falhas, mas também introduzem novos recursos assinados pela comunidade. Isso assegura que você tenha acesso às últimas melhorias de segurança;

6. **Suporte da Comunidade**: Existe uma forte comunidade de usuários e desenvolvedores ao redor do OPNsense, que podem oferecer suporte, recursos e documentação. Além disso, há opções de suporte comercial disponíveis, caso você precise de assistência profissional;

[3]É uma versão mais segura do HTTP, que utliza mecanismos de criptografia para assegurar o conteúdo da transmissão.

[4]O IDS (*Intrusion Detection System*) é um sistema de detecção de intrusões em redes que monitora o tráfego em busca de ameaças.

[5]O IPS (*Intrusion Prevention System*) é um sistema de segurança de rede que monitora e bloqueia ameaças.

[6]Uma VPN (*Virtual Private Network*) é uma rede privada virtual que permite a conexão entre dispositivos através da *Internet*, criando um túnel criptografado que protege os dados trafegados por ela.

7. **Flexibilidade**: OPNsense pode ser executado em uma variedade de *hardware*, de máquinas físicas a máquinas virtuais, o que permite fácil integração em ambientes já existentes;

8. **Compatibilidade e Integração**: OPNsense pode ser integrado a outras ferramentas de segurança e gerenciamento de rede, aumentando sua eficiência na detecção e resposta a ameaças;

9. **Desempenho**: Dependendo do *hardware* usado, o OPNsense pode proporcionar um desempenho superior, especialmente em ambientes que demandam altas taxas de transferência de dados.

Esses fatores fazem do OPNsense uma escolha atraente para muitos usuários em busca de uma solução robusta e flexível de NGFW, atendendo assim, aos requisitos estabelecidos pela equipe de TI do *campus*.

2.2 *Firewall*

Firewalls são elementos de *hardware* ou *software* que controlam o fluxo de tráfego entre redes ou dispositivos (*hosts*) e são importantes em cibersegurança. Eles podem ser de dois tipos:

- ***Firewalls* baseados em *host*:** São aqueles que ajudam a filtrar o tráfego para os dispositivos finais ou *hosts* [Mothersole e Reed 2011];

- ***Firewalls* de rede:** São aqueles que ajudam a fornecer segurança entre redes e são executados em *hardware* de rede.

Os *firewalls* de rede podem operar em diferentes camadas do modelo de Sistemas Abertos de Interconexão (*Open Systems Interconnection* – OSI), como a Camada de Transporte ou a Camada de Aplicação, filtrando pacotes de dados de acordo com regras predefinidas, cada um com características específicas para atender a diferentes necessidades de segurança, como por exemplo: *Packet filtering firewall*, *Stateful inspection firewalls*, *Application-level firewalls* e *Next-generation firewalls*.

2.2.1 *Packet filtering firewall*

Packet Filtering Firewall (*firewall* de filtragem de pacotes) é o mais antigo e mais básico tipo de *firewall* de rede. Os pacotes podem ser filtrados por endereços de rede de origem ou destino, protocolo, portas de origem ou destino.

Como a maioria das comunicações baseadas em Protocolo de Controle de Transmissão (*Transmission Control Protocol* — TCP) ou Protocolo de Datagrama do Usuário (*User*

Datagram Protocol — UDP) usa portas bem conhecidas para aplicações específicas, é conveniente bloquear pacotes com base em números de porta. Dessa forma, esse *firewall* não examina o conteúdo do pacote para tomar a decisão de permitir ou de rejeita-lo [Mothersole e Reed 2011].

Este tipo de *firewall* não armazena informação relacionada à conexão anterior, portanto, é chamado de *firewall* de filtro de pacote sem estado (*stateless firewall*), em que cada pacote é tratado como uma entidade individual, que é roteada se corresponder à lista de regras ou descartada se não corresponder.

Assim, a decisão de filtragem é feita separadamente para cada pacote, sem levar em consideração decisões anteriores para pacotes relacionados. *Firewalls* sem estado são rápidos e não mantêm registros históricos das conexões estabelecidas, pois processam pacotes de entrada e saída de forma independente e sem correlação de pacotes examinados anteriormente.

Por outro lado, para serviços bidirecionais, esses *firewalls* devem ter regras explicitamente escritas para permitir fluxos de tráfego de retorno, o que os torna expostos a ataques, como *spoofing*[7] e fragmentação [Trabelsi *et al.* 2018].

2.2.2 *Stateful inspection firewalls*

Stateful inspection firewalls (*firewalls* com inspeção de estado), são *firewalls* de estado que utilizam tabelas de sessão para manter o fluxo do estado das conexões.

A tabela de sessão, também conhecida como tabela de rastreamento de conexão (*connection track table*), é parte da estrutura interna do *firewall*. Ela rastreia conexões ativas e inspeciona pacotes para verificar se eles fazem parte de uma sessão ativa anterior ou não. Se os pacotes tiverem as propriedades esperadas que a tabela de sessão prevê, os pacotes são encaminhados sem processamento pelas regras de filtragem.

As informações relacionadas às entradas da tabela de sessão dependem do fornecedor do *firewall*. Mas, elas normalmente incluem os seguintes campos: Protocolo, Endereço IP de origem, Porta de Origem, Endereço IP de destino, Porta de destino, Estado da conexão e seus tempos limite. Um estado é rastreado até que uma conexão seja interrompida ou até que um tempo limite pré-configurado seja atingido.

Eles também podem incluir um número de sequência, um número de confirmação e um tamanho de janela. A principal vantagem do *firewall* com estado é que ele é mais seguro do que o *firewall* sem estado, pois o administrador não precisa escrever regras de filtragem para permitir o tráfego de retorno. Isso fecha a brecha de segurança aberta por regras que permitem o tráfego de retorno, que pode ser explorado por um invasor para lançar um ataque de negação de serviço (*Deny of Service* – DoS).

Além disso, a lista de regras no *firewall* com estado é mais curta, pois uma única regra é suficiente para descrever um fluxo (entrada e saída); enquanto duas regras são necessárias

[7] *spoofing* Técnica de falsificação em que elementos da rede, como endereços IP ou serviços, forjam endereços ou serviços legítimos para obter vantagens ou acessos que naturalmente não teriam.

para o mesmo fluxo no *firewall* sem estado correspondente [Trabelsi *et al.* 2018].

2.2.3 *Application-level firewalls*

Application-level firewalls (*firewalls* de nível de aplicação) é um *firewall* que opera na Camada de Aplicação do OSI e também é conhecido como *Proxy*[8] *Firewall*.

O *proxy firewall* estabelece a conexão com a fonte e então faz a inspeção do conteúdo do pacote e realiza a verificação da presença de *malware*.[9] Somente se os dados forem aprovados, eles são enviados ao destino. Assim, os servidores *proxy* adicionam uma camada de proteção ao fornecer anonimato aos dispositivos dentro da rede. Esses dispositivos *proxy* geralmente estão presentes na borda de uma rede. No entanto, pode haver uma desaceleração na taxa de transferência devido às medidas de segurança adicionais. Esta é uma das desvantagens desta tecnologia.

Um *proxy firewall* intercepta uma solicitação de um sistema na rede interna [Aziz *et al.* 2012] antes de enviá-la ao destino. Ele se comporta como um servidor ao se comunicar com o *host* cliente e como um cliente ao enviar ou receber dados do *host* servidor. Dessa maneira, o servidor e o cliente não apresentam uma conexão direta. A decisão de permitir ou negar a transmissão de dados é baseada nos cabeçalhos ou *payloads* (porção útil dos dados do usuário) do protocolo de aplicação. Assim, os *proxy firewalls* oferecem um maior nível de segurança.

2.2.4 *Next-generation firewalls*

Next-generation firewalls – NGFW (*firewalls* de próxima geração), diferente dos *firewalls* tradicionais (*stateful*), que são projetados apenas para filtrar tráfego baseado em portas, protocolos e estados de conexão, combinam métodos como inspeção profunda de pacotes (*Deep Packet Inspection* – DPI), sistema de prevenção de intrusão (*Intrusion Prevention System* – IPS), gerenciamento de largura de banda, filtragem de URL (*Uniform Resource Locator*), antivírus, detecção de *malware*.

Os NGFW podem evitar ameaças à rede e fornecer um nível mais alto de segurança. Não há uma definição fixa de um *firewall* de próxima geração. Eles inspecionam o conteúdo e sinalizam dados potencialmente prejudiciais usando métodos baseados em assinatura ou em aprendizado de máquina.

Como comparação, houve uma pesquisa conduzida para estudar a eficácia do *Firewall* de Próxima Geração em *Internet* das coisas (*Internet of Things* – IoT) para uma casa inteligente e uma empresa. O *firewall* é testado com vários ataques como Negação de Serviço

[8]Proxy: Serviço que age como um intermediário entre o usuário e a *Internet*, recebe e repassa todas as requisições ao site que o usuário está acessando.

[9]*Malware* é um termo genérico para qualquer tipo de *software* malicioso projetado para prejudicar ou explorar qualquer dispositivo, serviço ou rede programável.

Distribuída (*Distributed Deny-of-Service* – DDoS), *phishing* e SQL[10] *Injection* em diferentes redes. Os resultados mostraram que o *firewall* de próxima geração tem melhor desempenho na proteção contra ameaças, por possuir mecanismos de detecção e filtragem mais eficientes, quando comparado a tecnologias de *firewall* mais antigas [Soewito e Andhika 2019].

2.3 OPNsense

O OPNsense[11] é uma plataforma de *firewall* e roteamento baseada em FreeBSD[12] de código aberto, fácil de usar e de construir, por possuir uma Interface Gráfica do Usuário (*Graphical User Interface* – GUI) intuitiva que proporciona a configuração e gerência de seus recursos de forma ágil, quando comparada as soluções que necessitam de interações via linha de comando. Inclui a maioria dos recursos disponíveis em *firewalls* comerciais e benefícios de fontes abertas e verificáveis.

Teve seu início como um *fork*[13] do pfSense e do `m0n0wall` em 2014, com seu primeiro lançamento oficial em janeiro de 2015. O projeto evoluiu rapidamente, embora ainda mantenha aspectos familiares do m0n0wall e do pfSense. Um forte foco em segurança e qualidade de código impulsiona o desenvolvimento do projeto.

O OPNsense oferece atualizações de segurança semanais com incrementos para reagir a novas ameaças emergentes. Possui um ciclo fixo de dois lançamentos principais e a cada ano oferece às empresas a oportunidade de planejar atualizações com antecedência. Para cada lançamento principal, um roteiro é colocado em prática para orientar o desenvolvimento e definir metas claras [OPNSense 2024].

2.4 O iptables

O projeto iptables, também conhecido como `netfilter/iptables`, é um programa de *firewall* baseado em linha de comando para o sistema operacional *Linux* [netfilter 2025]. Ele utiliza tabelas para gerenciar e controlar o tráfego de rede. Suas principais características incluem:

1. **Filtragem de pacotes:** Permite aceitar (*accept)*, rejeitar (*reject*) ou descartar (*drop*) pacotes de acordo com critérios como IP de origem, IP de destino, protocolo e porta;

[10] SQL ou *Structured query language* é uma linguagem para criação, manipulação e gerenciamento de bancos de dados.

[11] (Disponível em https://opnsense.org/download/)

[12] O FreeBSD É um sistema operacional *Unix* baseado no sistema operacional BSD (*Berkeley Software Distribution*).

[13] *fork* Em engenharia de *software*, uma bifurcação ou ramificação acontece quando um desenvolvedor inicia um projeto independente com base no código de um projeto já existente, ou seja, quando um *software* é desenvolvido com base em outro, já existente, sem a descontinuidade deste último.

2. **Tradução de Endereços de Rede (NAT):** Realiza tarefas de NAT, como mascaramento (*masquerade*), DNAT (*Destination* NAT), SNAT (*Source* NAT) e redirecionamento (*redirect*);

3. **Marcação de tráfego:** Realiza a marcação (identificação) para manipulação de tráfego de rede;

4. **Controle de Encaminhamento:** Gerencia pacotes que passam por diferentes interfaces de rede;

5. **Registros e Auditorias:** Possibilita o registro (LOG) de pacotes para monitoramento e auditoria.

Como estrutura básica para o gerenciamento do tráfego, o `iptables` possui:

1. **Tabelas**

   - **FILTER:** tabela padrão que é responsável pela filtragem de pacotes;
   - **NAT:** tabela que permite modificar os endereços IP e portas dos pacotes;
   - **MANGLE:** tabela usada para modificar os cabeçalhos dos pacotes;
   - **RAW:** tabela usada para a manipulação de pacotes antes do rastreamento de conexões.

2. ***Chains***

   - **INPUT:** Processa pacotes destinados ao sistema local;
   - **OUTPUT:** Processa pacotes originados no sistema local para saída;
   - **FORWARD:** Processa pacotes encaminhados através das interfaces do sistema local (roteamento);
   - **PREROUTING:** Processa pacotes antes do processo de roteamento (usado no NAT e no MANGLE);
   - **POSTROUTING:** Processa pacotes depois do processo de roteamento (usado no NAT e no MANGLE);

2.5 Alta disponibilidade (*High Availability* - HA)

A alta disponibilidade (*high availability* – HA) é a capacidade de um sistema continuar operando sem interrupções, mesmo diante de falhas, manutenções ou problemas técnicos. Aplicar essa técnica aos *firewalls*, possibilita eliminar os pontos únicos de falha e garantir que a proteção da rede seja constante e ininterrupta, sem elementos que possam comprometer a cibersegurança. A perda temporária de um *firewall* pode deixar uma rede vulnerável a

ataques, o que pode resultar em prejuízos financeiros significativos, danos à reputação e perda de dados críticos [Godoy 2016]. Para promover a alta disponibilidade, são necessários os seguintes elementos chave:

- Redundância: Consiste em configurar dois ou mais *firewalls* (geralmente primário e secundário) de maneira que, em caso de falha no *firewall* principal, o secundário assuma automaticamente. Tipicamente implementada em uma configuração ativo-passivo (um *firewall* opera enquanto o outro está em espera) ou ativo-ativo (ambos operam simultaneamente, balanceando a carga).

- *Failover*[14] automático: É um mecanismo que detecta falhas no *firewall* ativo e realiza a transição do tráfego para o secundário de maneira transparente para os usuários, minimizando ou anulando o tempo de inatividade.

- Sincronização de configurações e estados: Em *firewalls stateful*, a sincronização é crítica para garantir que o *firewall* secundário mantenha o estado das conexões existentes, permitindo continuidade sem desconexões quando ocorre a mudança entre as unidades ativas. Configurações, políticas de segurança e regras também precisam estar atualizadas entre as unidades redundantes.

O OPNsense utiliza o Protocolo de Redundância de Endereço Comum (*Common Address Redundancy Protocol* – CARP) para *failover* de *hardware*. Dois ou mais *firewalls* podem ser configurados como um grupo de *failover*, se uma interface falhar no primário ou o primário ficar *offline* completamente, o secundário se torna ativo.

A utilização desse recurso no OPNsense, cria um *firewall* totalmente redundante com *failover* automático e contínuo. Ao alternar para o *backup*, as conexões de rede permanecerão ativas com interrupção mínima para os usuários.

O CARP foi criado por desenvolvedores do OpenBSD como uma solução de redundância aberta e gratuita para compartilhar endereços IP entre um grupo de dispositivos de rede. Foi desenvolvido como alternativa ao Protocolo de Redundância de Roteador Virtual (*Virtual Router Redundancy Protocol* – VRRP) e tornou-se disponível em outubro de 2003 no OpenBSD, e mais tarde foi adicionado ao FreeBSD também.

Ele utiliza pacotes *multicast* para sinalizar seus vizinhos sobre seu *status*. Cada interface virtual deve ter um *Virtual Host ID* (**vhid**) exclusivo, que é compartilhado entre as máquinas do grupo *failover*. Para determinar qual máquina desse grupo tem uma prioridade mais alta, o *skew*[15] anunciado é usado. Um *skew* menor implica uma prioridade mais alta (o *firewall* principal usa **0**).

[14] *Failorver* ou tolerância a falha se dá quando um sistema, servidor ou outro componente de *hardware* ou *software* fica indisponível e um componente secundário assume operações sem que haja interrupção nos serviços.

[15] *skew* é o intervalo de tempo usado para compensar diferenças nos tempos de vida de pacotes entre os dispositivos no grupo, ajudando a evitar falhas quando o roteador primário não responde.

Junto ao CARP, o OPNsense usa os seguintes recursos:

- pfSync: É usado para replicar o estado das conexões dos *firewalls*. Ao fazer *failover*, é necessário que os *firewalls* tenham registradas todas as conexões para tornar a migração perfeita. Isso é garantido por meio da configuração de uma interface entre os dispositivos de *firewall*, preferencialmente, uma interface dedicada por questões de segurança em desempenho.
- XMLRPC *sync*: É um mecanismo para manter a configuração do servidor de *backup* em sincronismo com o principal e é chamado de sincronização XMLRPC [Docs-OPNSense 2024].

A arquitetura utilizada no CARP pode ser observada na Figura 2.1.

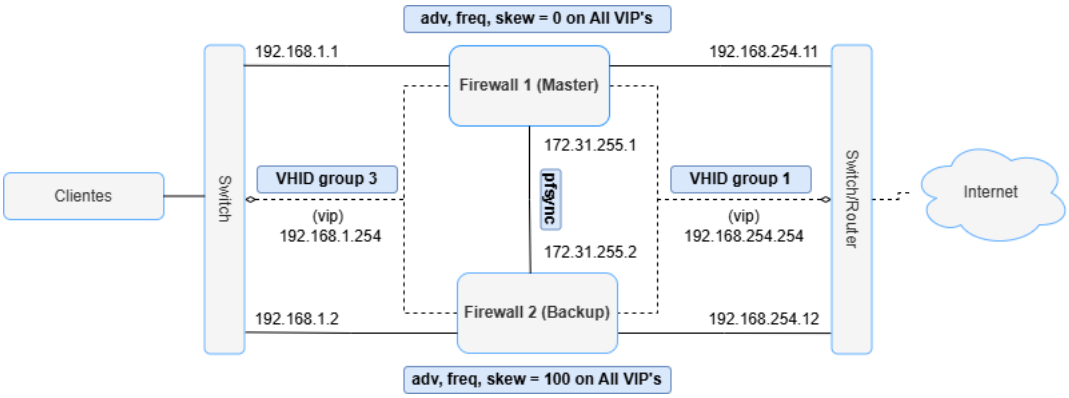


Figura 2.1: *Exemplo da arquitetura utilizada no CARP*
Fonte: Autoria própria

2.6 PNETLab

O PNETLab é uma ferramenta voltada para a simulação de ambientes de redes por meio da virtualização de sistemas operacionais. Com ele é possível criar e compartilhar laboratórios com a comunidade [PNETLab 2024].

O papel dessa ferramenta nesse trabalho é simular, testar e validar o ambiente sugerido, minimizando a possibilidade de erros e prevendo possíveis problemas na implementação da proposta. Além disso, possibilita visualizar de forma geral e simplificada a topologia e arquitetura utilizada.

2.7 LAGG – (*Link Aggregation*)

O LAGG é uma técnica de rede utilizada para combinar múltiplos *links* físicos entre dispositivos de rede (como *switches*, servidores ou roteadores) em uma única conexão lógica, aumentando a largura de banda e proporcionando redundância. O objetivo do LAGG é

melhorar o desempenho de rede, distribuir o tráfego entre os *links* agregados de forma balanceada e garantir maior confiabilidade em caso de falha de um deles [IEEE 2025].

Embora o LAGG seja o conceito de agregação de *links*, ele frequentemente é implementado com protocolos, como o LACP (*Link Aggregation Control Protocol*) ou *Static Link Aggregation*, que ajudam na negociação e no gerenciamento das agregações.

2.7.1 LACP – (*Link Aggregation Control Protocol*)

O LACP é um protocolo de rede definido no padrão IEEE 802.1AX (anteriormente IEEE 802.3ad) que permite a agregação de múltiplos *links* físicos em uma única conexão lógica, assim como o LAGG, no entanto, ele pode realizar a negociação automática entre os dispositivos de rede para formar um *link* agregado (LAG - *Link Aggregation Group*).

O protocolo possui, inclusive, o monitoramento continuo do estado dos *links* individuais, ajustando a agregação conforme necessário, e permite a remoção ou adição dinâmica de *links* sem interrupções significativas no tráfego. Suas principais características são:

- Negociação automática: O LACP permite que os dispositivos de rede detectem e negociem a agregação de *links* de forma automática, sem necessidade de configuração manual detalhada.
- Redundância e alta disponibilidade: Em caso de falha de um dos *links* agregados, o LACP redireciona o tráfego automaticamente para os *links* restantes, garantindo a continuidade da comunicação.
- Balanceamento de carga: O LACP pode distribuir o tráfego entre os *links* agregados com base em critérios como endereço MAC, endereço IP, ou o número de pacotes transmitidos, otimizando o uso dos *links*.
- Compatibilidade e interoperabilidade: LACP é suportado por diversos fabricantes de dispositivos de rede, o que facilita a integração de diferentes dispositivos de rede no mesmo link agregado.

2.8 VLAN – (*Virtual Local Area Network*)

VLAN é uma técnica de segmentação de redes que permite a criação de redes lógicas independentes, dentro de uma mesma infraestrutura física. Com as VLANs, os dispositivos que pertencem a uma mesma rede lógica podem se comunicar entre si, mesmo que estejam fisicamente dispersos em diferentes locais ou dispositivos. Elas são amplamente usadas para melhorar a segurança, o desempenho e a gestão da rede.

As VLANs operam no nível da camada de enlace (camada 2) do modelo OSI e são definidas por identificadores chamados de VLAN IDs. Cada VLAN tem um número único associado a ela, o que permite que dispositivos membros de uma mesma se comuniquem

sem interferir com dispositivos de outras, mesmo que compartilhem o mesmo *switch* ou infraestrutura de rede física.

Equipamentos de rede como *switches*, são configurados para identificar o tráfego de cada VLAN por meio da marcação de pacotes (*tagging*) usando o protocolo 802.1Q, que adiciona um campo de VLAN ID no quadro *Ethernet*.

Como o tráfego entre VLANs é logicamente isolado, para permitir a comunicação entre VLANs diferentes, é necessário um roteador ou um *switch multilayer* (camada 3), que pode realizar o roteamento entre as VLANs.

As principais características e benefícios das VLANs são:

- Segurança: Ao separar grupos de dispositivos em VLANs distintas, pode-se restringir a comunicação entre eles, garantindo que apenas usuários ou dispositivos autorizados em uma VLAN possam se comunicar diretamente.
- Melhoria no desempenho: Podem reduzir o tráfego desnecessário em uma rede, pois as comunicações são limitadas àqueles dispositivos que estão na mesma VLAN. Isso pode diminuir o congestionamento da rede.
- Facilidade de gerenciamento: Com elas, é possível gerenciar a rede de forma mais eficiente, criando segmentos lógicos que podem ser configurados ou reconfigurados sem mudanças físicas na rede.
- Flexibilidade: Permitem que dispositivos em diferentes localizações físicas (como diferentes andares ou prédios) se comuniquem como se estivessem na mesma rede local.

2.9 *Virtual Machine* – VM

Máquina Virtual (*Virtual Machine* – VM) é um computador digitalizado, que simula o funcionamento de um computador físico, permitindo que múltiplos sistemas operacionais e aplicativos sejam executados em um único *hardware* físico de forma isolada. As máquinas virtuais compartilham os recursos físicos do *host*, como CPU, memória e disco, mas operam de forma independente, como se fossem máquinas separadas [RedHat 2025].

Os principais componentes de uma VM são:

1. **Hipervisor** (*Hypervisor*): *Software* que cria e gerencia as máquinas virtuais, dividindo os recursos do *hardware* entre elas, basicamente podem ser divididos em dois tipos:
   - **Hipervisor de tipo 1**: Também chamado de *bare-metal*,é executado diretamente no *hardware* físico (ex.: KVM, Proxmox, VMware ESXi, Microsoft Hyper-V) [RedHat 2025];
   - **Hipervisor de tipo 2**: Executado sobre um sistema operacional (*hosted* –hospedado) (ex.: *Oracle VirtualBox*, VMware *Workstation*) [RedHat 2025];

2. **Sistema Operacional Convidado (*Guest* OS)**: O sistema operacional instalado dentro da máquina virtual;

3. **Imagens Virtuais de Disco**: Arquivos que simulam discos rígidos usados para armazenar o sistema operacional e dados.

2.10 Protocolo ICMP

O ICMP (*Internet Control Message Protocol*), é um protocolo da camada de rede do modelo OSI (camada 3), que faz parte da pilha de protocolos TCP/IP. Sua principal função é o envio de mensagens de controle, diagnóstico e erro entre dispositivos na rede, como roteadores e *hosts*. O ICMP é essencial para a operação de redes IP, pois permite a comunicação sobre problemas de transmissão de pacotes e fornece ferramentas de diagnóstico, como o comando ping [Kurose e Ross 2017].

As principais funções do ICMP são:

1. **Mensagens de erro:** O ICMP envia notificações sobre falhas de entrega de pacotes, como quando o destino está inacessível (*Destination Unreachable*), ou o tempo de vida (TTL) de um pacote expirou (*Time Exceeded*);

2. **Mensagens de controle e diagnóstico:** Permite verificar a conectividade de rede com os pacotes *Echo Request* e *Echo Reply*, usados em ferramentas como o ping, para testar se um *host* está acessível;

3. **Controle de tráfego:** Em situações de congestionamento, o ICMP pode usar a mensagem *Source Quench* para pedir ao remetente que desacelere o envio de pacotes.

Uma das aplicações para o protocolo ICMP, é a detecção de perda de pacotes, que é um fenômeno em redes de computadores onde um ou mais pacotes de dados não chegam ao destino esperado. Isso pode ocorrer devido a diversos fatores, como congestionamento na rede, falhas de *hardware*, problemas de roteamento, ou limitações de largura de banda.

A perda de pacotes é uma das principais causas de degradação de desempenho em redes de comunicação, especialmente em protocolos que não possuem mecanismos para garantir a entrega de dados (como o UDP) e serviços como VoIP, videoconferências, jogos online, entre outros [Cardellini *et al.* 2010].

Capítulo 3

Metodologia

Este trabalho tem como objetivo o planejamento para migração da solução legada de *firewall* do IFPB *campus* Campina Grande, para uma solução utilizando o OPNsense em alta disponibilidade.

Em entrevistas realizadas com o responsável técnico e em visitas ao setor de TI da instituição, o cenário atual da estrutura de redes e de *firewall* utilizados no *campus* foi apresentado. O sistema de *firewall* atual é antigo e funciona em uma máquina virtual com o sistema operacional *Debian*, versão 5 que utiliza o pacote `netfilter/iptables`.

A configuração atual dessa VM conta com 2 *gigabytes* de memória RAM, 36 *gigabytes* de armazenamento e 4 vCPUs (CPU virtual), conforme Figura 3.1.

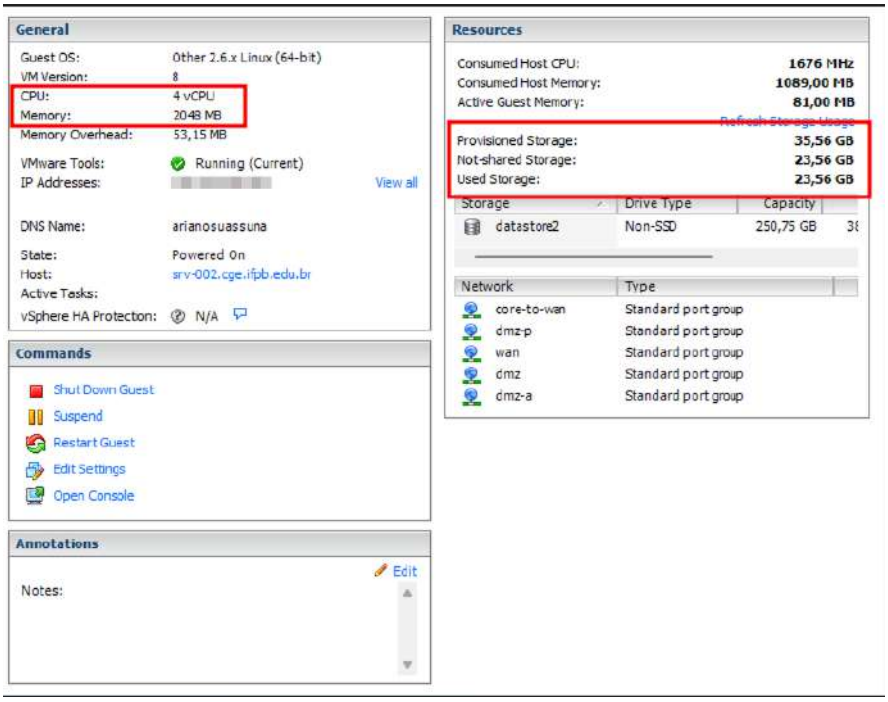


Figura 3.1: *Recursos da VM de firewall atual*

A definição das regras de filtragem de *firewall* e tradução de endereços de rede (*Network Address Translation* – NAT) é realizada a partir de um *script*, configurado pre-

viamente, que é executado durante a inicialização do sistema operacional. A partir da sua execução, todas as políticas de acesso, filtragem e tradução necessárias são definidas.

Os ajustes nas políticas ou regras são realizados por meio da edição desse *script*, isso faz o trabalho por parte do administrador ficar mais lento e aumenta a possibilidade de problemas, pois, qualquer erro de digitação pode fazer com que o *script* não seja executado corretamente. Com isso, o sistema de *firewall* não é ativado e a rede fica vulnerável à ameaças.

Para a elicitação dos requisitos necessários para a implantação da nova solução, foi elaborado um questionário (presente no apêndice A) contendo a pesquisa com as principais demandas do setor. A partir desse levantamento foi possível planejar uma topologia e um ambiente mais alinhado às necessidades de segurança, desempenho e disponibilidade da instituição.

Tendo em vista a sustentabilidade e a necessidade de alocação de recursos financeiros, a utilização de uma ferramenta *open source* e gratuita foi escolhida, assim, a viabilização do projeto ficou mais simples, sem comprometer a robustez e segurança do mesmo. Uma pesquisa para a escolha da ferramenta que atenda as necessidades do setor e da instituição foi realizada, assim as principais características foram levantadas, como pode ser observardo na Tabela 3.1:

Tabela 3.1: *Comparação entre OPNsense, pfSense e iptables*

Critério	**OPNsense**	**pfSense**	**iptables**
Licença	*Open Source* (BSD)	*Open Source* (BSD)	*Open Source* (GPL)
Interface de Gerenciamento	Interface *Web* com *design* moderno	Interface *Web* (baseada em FreeBSD)	Linha de Comando (CLI), GUI opcional
Base do Sistema Operacional	FreeBSD	FreeBSD	Linux
Requisitos de *Hardware*	Consome poucos recursos, maior consumo que iptables	Similar ao OPNsense	Leve, compatível com qualquer Linux
Facilidade de Uso	Muito fácil de usar	Fácil de usar	Complexo, exige conhecimento avançado
Suporte a Alta Disponibilidade (HA)	Suporte nativo a CARP, *failover* e sincronização	Suporte a CARP, *failover* e sincronização	Manual com *Keepalived* ou *Pacemaker*
Suporte a VPN	*Open*VPN, IPsec, *WireGuard*, PPTP	OpenVPN, IPsec, *WireGuard*, PPTP	Dependente de ferramentas externas
Inspeção de Pacotes (*Stateful*)	Sim	Sim	Sim
IDS/IPS	Suricata integrada	Snort ou Suricata	Configuração manual (Suricata, Snort)
Suporte a NAT	Completo	Completo	Completo
Sistema de Gerenciamento de Pacotes	Sistema de *plugins* nativo	Sistema de pacotes e *plugins*	Não possui sistema de pacotes nativo
Custo de Licenciamento	Gratuito	Gratuito	Gratuito
Documentação e Comunidade	Boa, ativa e em crescimento	Boa e ampla	Extensa, mas dispersa
Atualizações de Segurança	Atualizações regulares, com foco em estabilidade e segurança	Atualizações regulares	Dependente da distribuição *Linux*
Capacidade de Automação	Suporte a *scripts* e APIs	Suporte a APIs e automação	*Scripts* em *shell* e ferramentas externas

Levantadas essas informações, apresentadas na Tabela 3.1, pode-se justificar a escolha do OPNSense em detrimento ao pfSense (principal concorrente de código aberto baseado em FreeBSD/Unix), por conhecimento prévio da solução pela equipe de TI do *campus*, o que garante um melhor aproveitamento dos recursos e dispensa a necessidade de treinamentos específicos, e também, pelos seguintes aspectos:

1. **Modelo de desenvolvimento:**
   - **OPNsense**: Tem um foco em atualizações regulares e uma abordagem mais ágil e transparente para desenvolvimento. Os mantenedores do OPNsense frequentemente priorizam *feedback* da comunidade e melhorias na interface;
   - **pfSense**: Embora também atualize regularmente, as mudanças podem ser mais conservadoras devido à base de usuários maior e a complexidade do sistema.
2. **Interface de Usuário:**
   - **OPNsense**: Possui uma interface moderna e intuitiva, com um *design* que facilita a navegação e configurações mais simples para usuários menos experientes;
   - **pfSense**: A interface é funcional, mas pode parecer um pouco ultrapassada e menos amigável para novos usuários.
3. **Licenciamento e Comunidade:**
   - **OPNsense**: É *open source* e adota uma abordagem de transparência, onde todos os recursos e melhorias são compartilhados com a comunidade;
   - **pfSense**: Embora existam versões *open source*, também oferece uma versão com custos para aquisição (pfSense *Plus*), que vem com funcionalidades adicionais e suporte, o que pode limitar a transparência em alguns aspectos.
4. **Recursos e Funcionalidades:**
   - **OPNsense**: Frequentemente introduz novos recursos de forma mais rápida que o pfSense. Por exemplo, ele pode oferecer melhor suporte para integração de *plugins* e personalização;
   - **pfSense**: Já possui um ecossistema robusto de *plugins*, mas a adição de novos recursos pode ser mais cautelosa.
5. **Gestão de *Plugins*:**
   - **OPNsense**: Tem uma maneira mais simplificada e controlada de gerenciar *plugins*, com um repositório mais integrado;
   - **pfSense**: Suporta muitos *plugins*, mas sua gestão pode ser mais segmentada.

6. **Relatórios e Monitoramento:**

   - **OPNsense**: Inclui ferramentas de monitoramento e relatórios mais integradas e com visualização moderna, facilitando a análise de tráfego e eventos de segurança;
   - **pfSense**: Embora tenha opções de monitoramento, pode exigir ferramentas externas ou *plugins* adicionais para recursos mais avançados.

7. **Comunidade e Suporte:**

   - **OPNsense**: Tem uma crescente comunidade ativa e opções de suporte comercial. O envolvimento da comunidade é bem enfatizado;
   - **pfSense**: Tem uma base de usuários maior e uma comunidade estabelecida, mas o suporte comercial está centrado nas versões com custos de aquisição.

3.1 Topologia e arquitetura do cenário atual

A infraestrutura atual de redes do *campus*, possui a topologia e arquitetura como pode ser observada na Figura 3.2. Nesta, a rede é segmentada basicamente em 4 partes: administração, educação, *data center* e DMZs. Essas redes agregam seus respectivos serviços dentro do *campus*.

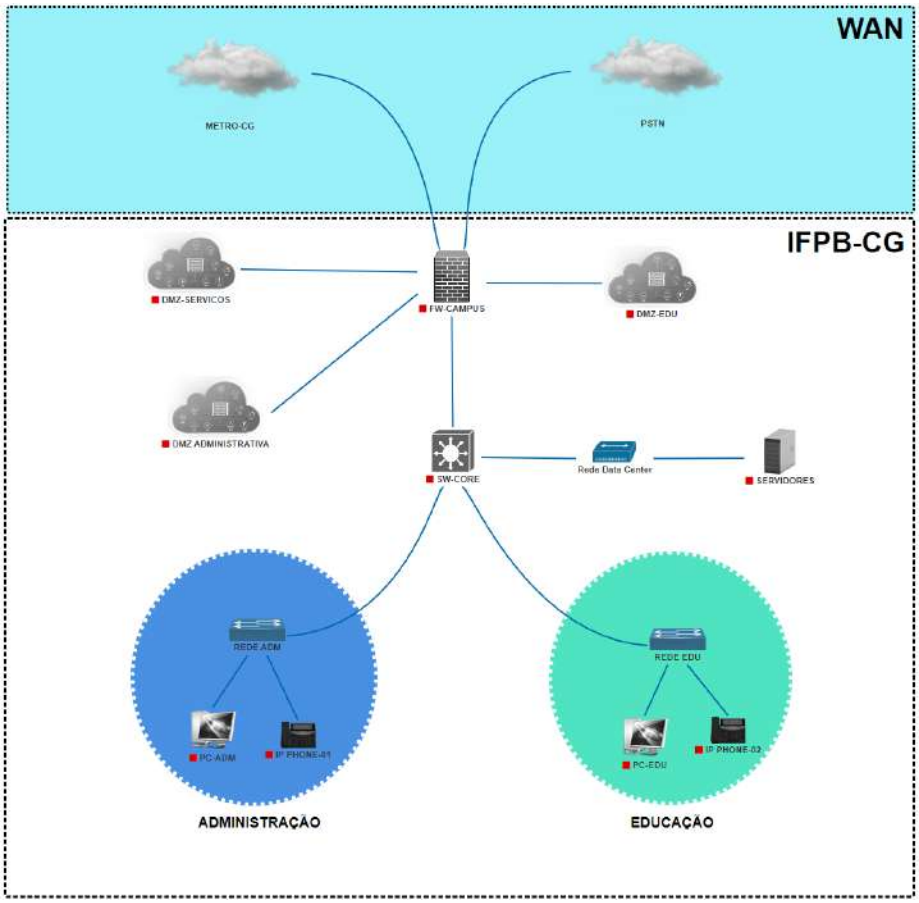


Figura 3.2: *Topologia e arquitetura atual*

- **DMZs:** Segmentos de rede que hospedam os serviços disponíveis também na *Internet*;
- **DATA CENTER:** Segmento de rede que hospeda os serviços internos do data center;
- **ADMINISTRAÇÃO:** Segmento de rede interna reservado para dispositivos da rede administrativa, como impressoras, computadores demais serviços de redes administrativos;
- **EDUCAÇÃO:** Segmento de rede interna reservado para os dispositivos da rede de educação ou acadêmica, como computadores dos laboratórios de informática e demais serviços de redes oferecidos aos alunos.

3.2 Topologia e arquitetura proposta

A topologia foi montada de acordo com o planejamento inicial dentro do ambiente do PNETLab para que fosse possível a execução das configurações e testes do ambiente. Por motivos de segurança e sigilo, foram utilizados dados genéricos para a configuração das redes e seus recursos neste trabalho.

A topologia e arquitetura proposta, pode ser observada na Figura 3.3. Cada *firewall* possui uma interface física dedicada, pela qual se estabelece a comunicação de sincronismo das funcionalidades de HA e configurações de cada *firewall*, além de duas interfaces de rede agregadas (LAGG) ligadas ao *switch* principal da rede (*switch core*). O tráfego de entrada (INBOUND) e de saída (OUTBOUND) da rede do *campus*, passa através dessas interfaces, possibilitando assim, um maior desempenho e tolerância a falha, pois, caso uma das interfaces apresente falha, a outra assume o tráfego sem interrupção dos serviços de redes oferecido pelos *firewalls*.

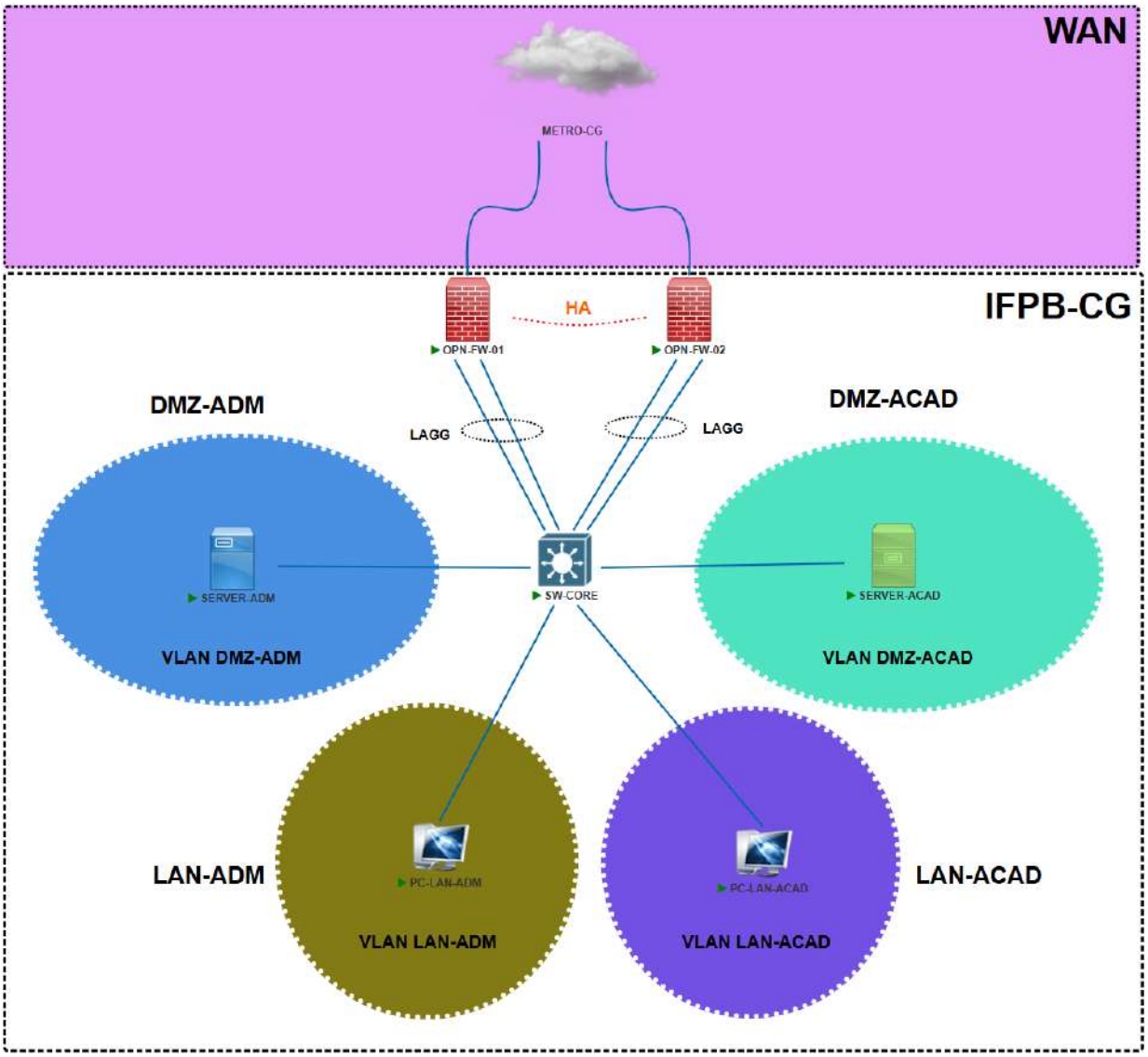


Figura 3.3: *Topologia e arquitetura da solução proposta.*

Entendendo a topologia proposta

A segmentação é feita com o uso de VLANs que estão presentes em cada interface agregada entre os *firewalls* e o *switch* principal da rede. A partir desse *switch*, as redes já segmentadas por VLANs, são configuradas nas suas interfaces para atender aos serviços presentes na infraestrutura da rede do *campus*.

- **WAN:** Segmento de rede entre os dispositivos de *firewall* do *campus* e a rede metropolitana de Campina Grande (METRO-CG);
- **HA:** Interface dedicada para a sincronização dos estados de HA e configurações entre os *firewalls*;
- **DMZ-ADM:** Segmento de rede onde os servidores e serviços administrativos estão disponíveis para a *Internet* e redes locais do *campus*;
- **DMZ-ACAD:** Segmento de rede onde os servidores e serviços acadêmicos estão disponíveis para a *Internet* e redes locais do *campus*;
- **LAN-ADM:** Segmento de rede interna reservado para dispositivos da rede administrativa, como impressoras, computadores demais serviços de redes;
- **LAN-ACAD:** Segmento de rede interna reservado para os dispositivos da rede acadêmica, como computadores dos laboratórios de informática e demais serviços de redes oferecidos aos alunos.

3.3 Instalação OPNsense

O *download* da imagem de instalação (ISO) foi efetuado no site oficial do OPNsense, e também, os requisitos de *hardware* recomendados para o provisionamento das VMs foram consultados, como apresentados na Tabela 3.2 [Docs-OPNSense 2024]:

Tabela 3.2: *Pré-requisitos de Hardware recomendados.*

Processador	1,5 Ghz *Multi core* CPU
RAM	8 *gigabytes*
Disco	80 *gigabytes*

- **Processador:** Processador com uma frequência de operação mínima de 1,5 Ghz e múltiplos núcleos de processamento;
- **RAM:** Quantidade mínima de 8 *gigabytes* de memória recomendada;
- **Disco:** Um disco com 80 *gigabytes* de espaço livre para instalação do sistema.

O processo de instalação do OPNsense foi dividido em passos, como pode se observar nas Figuras de 3.4 a 3.13 a seguir:

Passo 1: Iniciar sistema a partir da imagem (ISO) do OPNsense para dar início a instalação:

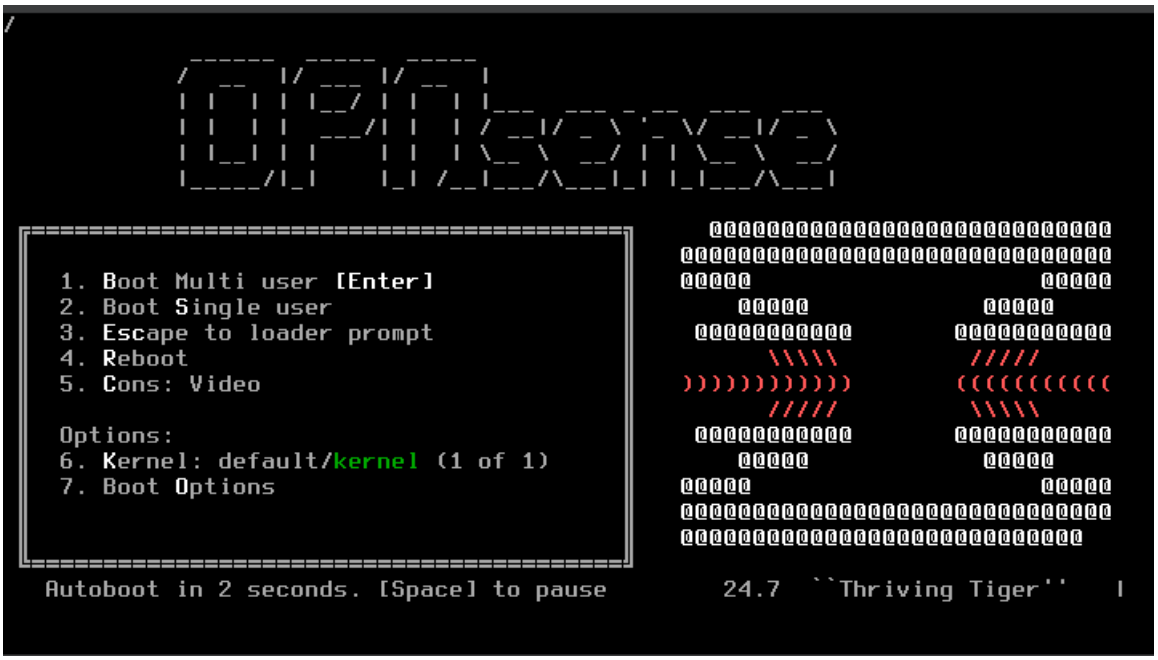


Figura 3.4: *Instalação OPNsense - Passo 1.*

Passo 2: Aguardar o processo de inicialização até chegar na área de *login*, em seguida, usar os dados para *login*:

- **Usuário:** *root*
- **Senha:** *opnsense*

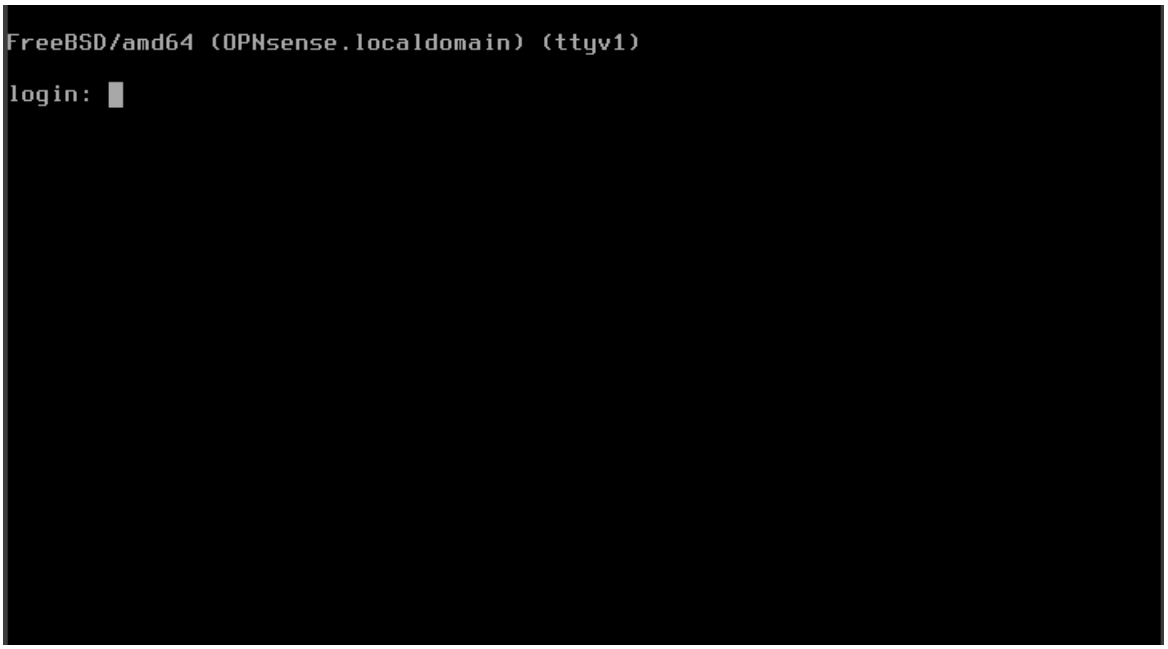


Figura 3.5: *Instalação OPNsense - Passo 2.*

Passo 3: Depois de efetuado o processo de *login*, digitar 8 como opção e pressionar a tecla *Enter* para ter acesso ao *prompt* de comandos.

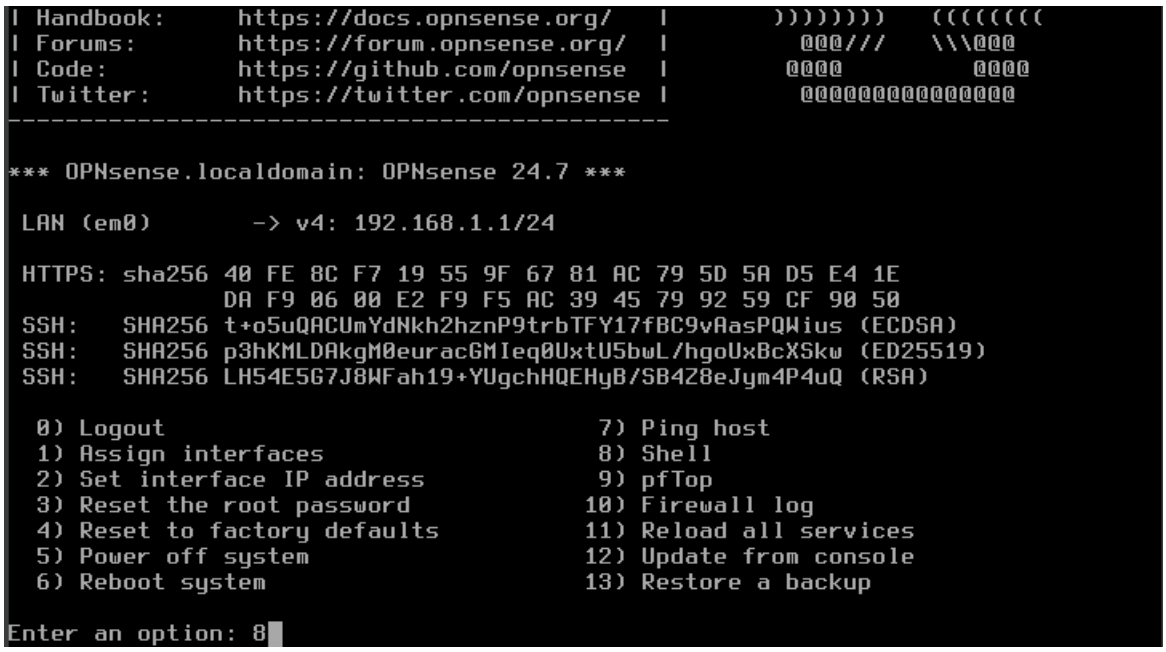


Figura 3.6: *Instalação OPNsense - Passo 3.*

Passo 4: No *prompt* de comando, digitar *opnsense-installer* e pressionar a tecla *Enter* para dar inicio à instalação do OPNsense no disco da VM.

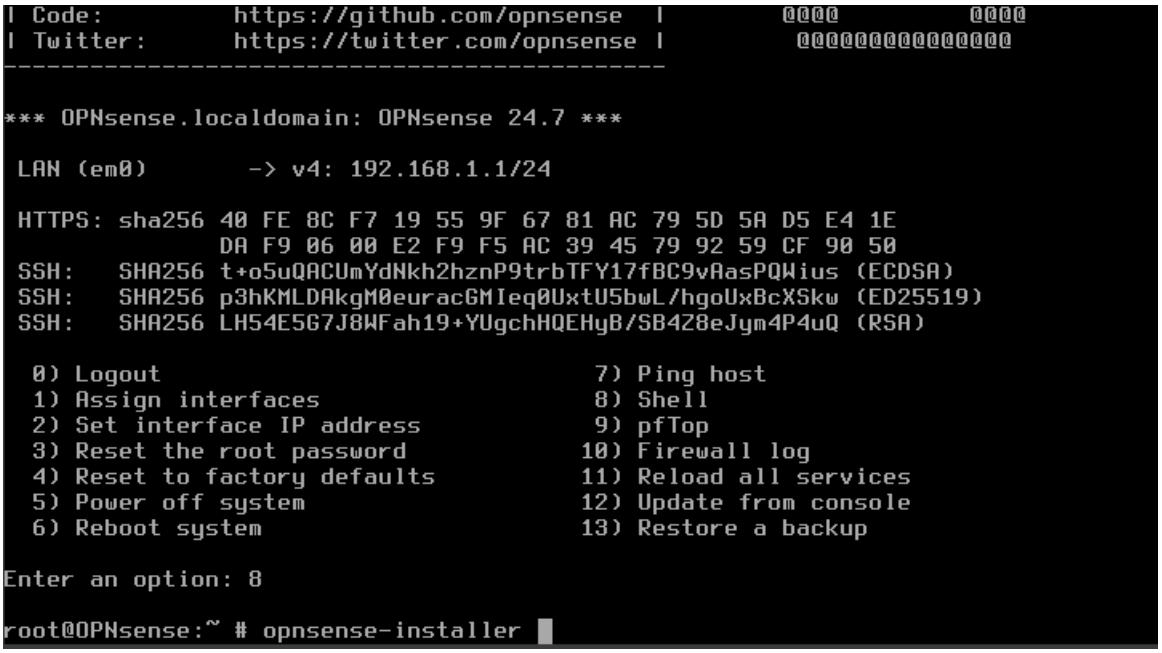


Figura 3.7: *Instalação OPNsense - Passo 4.*

Passo 5: Selecionar o leiaute de teclado apropriado e prosseguir.

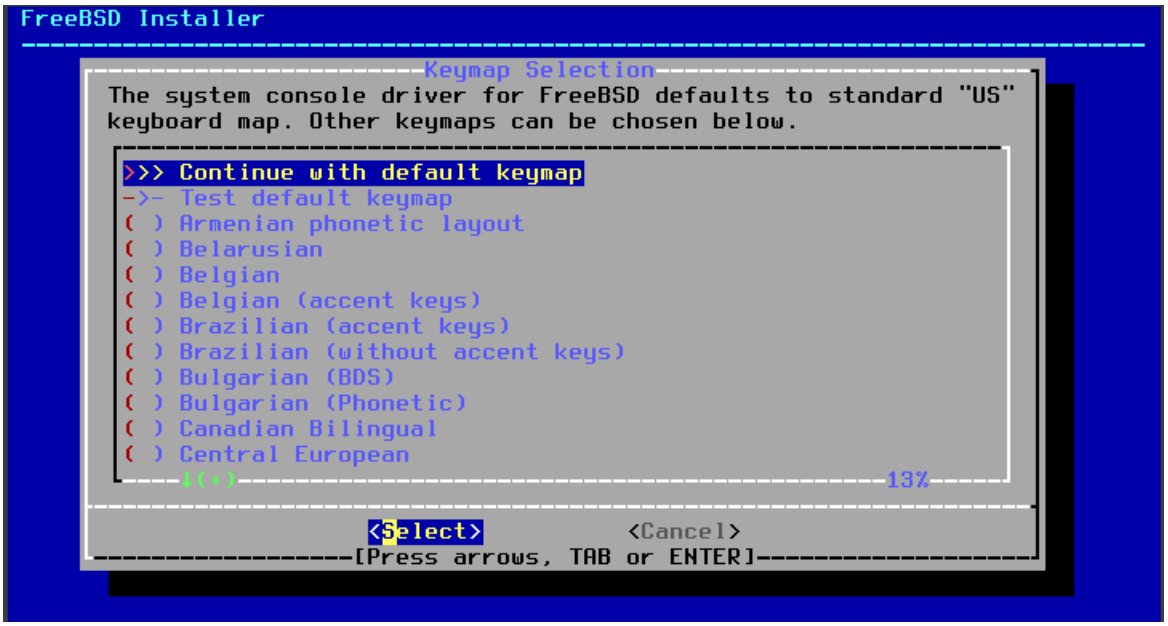


Figura 3.8: *Instalação OPNsense - Passo 5.*

Passo 6: Selecionar sistema de arquivos ZFS que será utilizado na formatação do disco da VM e instalação.

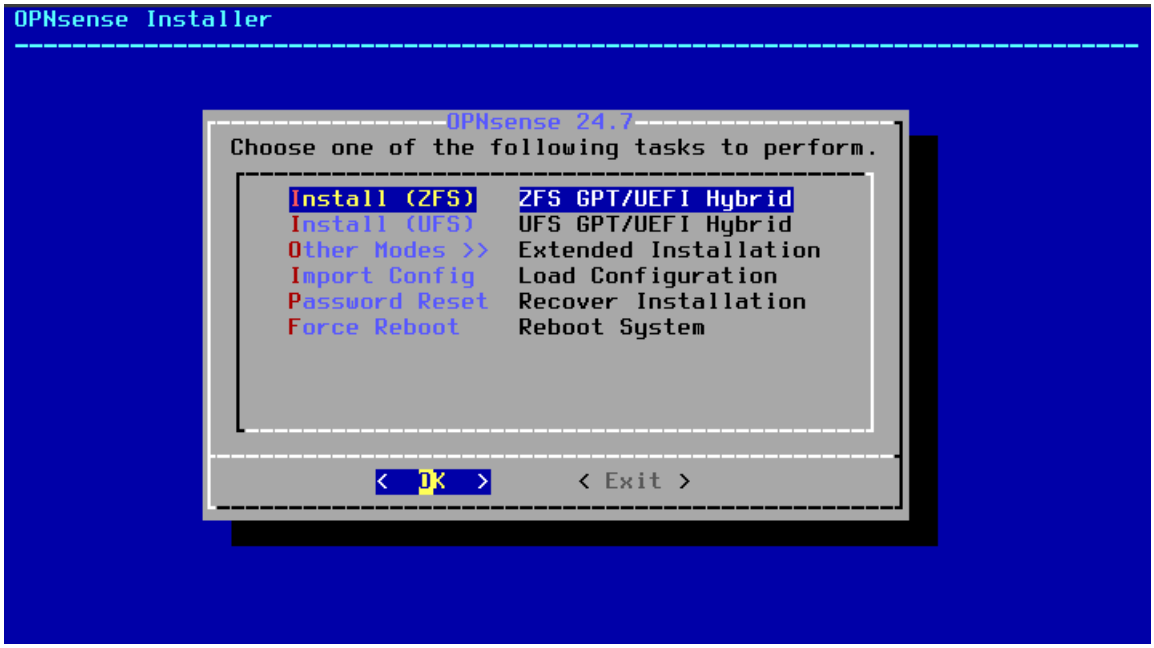


Figura 3.9: *Instalação OPNsense - Passo 6.*

Passo 7: Selecionar o tipo de matriz de discos (RAID[1]) que será utilizado.

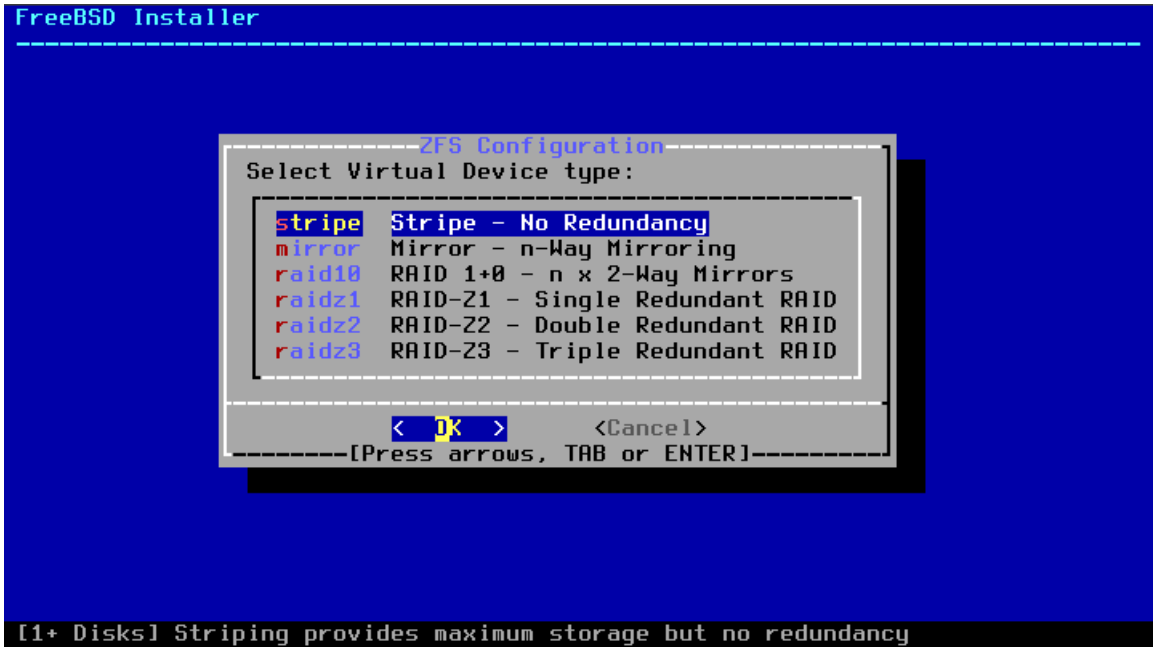


Figura 3.10: *Instalação OPNsense - Passo 7.*

Passo 8: Selecionar unidade de disco para a formatação e instalação.

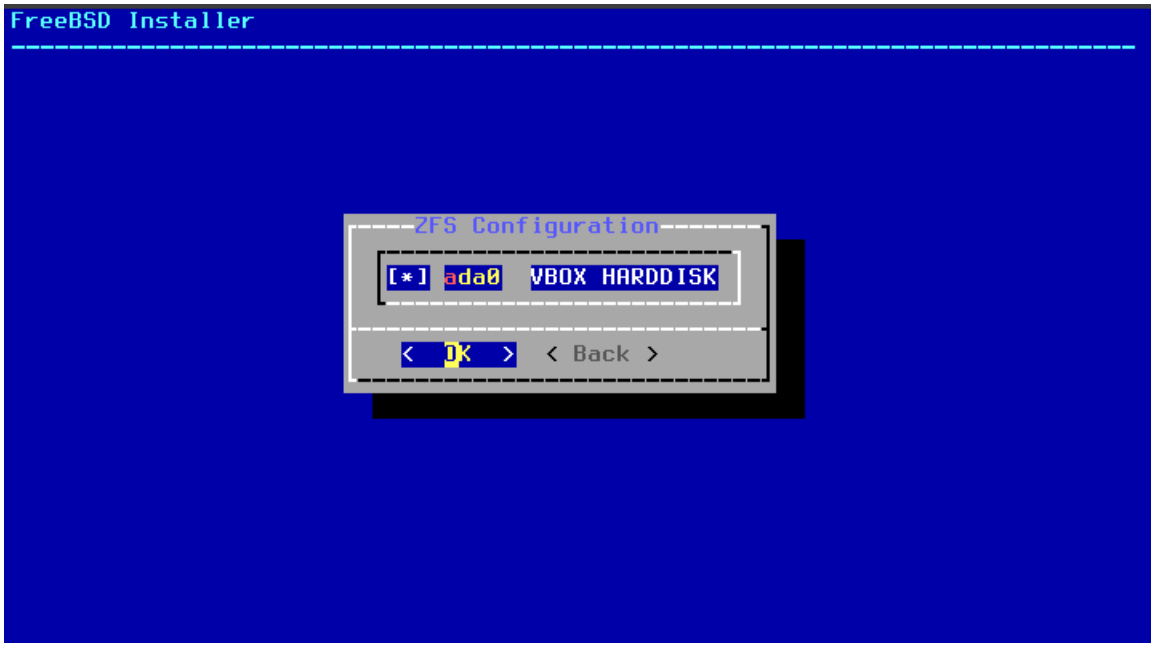


Figura 3.11: *Instalação OPNsense - Passo 8.*

[1]RAID é a sigla para *Redundant Array of Independent Disks*, que significa Matriz Redundante de Discos Independentes. É uma tecnologia de armazenamento que combina dois ou mais discos rígidos (HDDs) ou *solid-state drives* (SSDs) para formar uma única unidade lógica

Passo 9: Ao selecionar o disco que será particionado, formatado e instalado o sistema, será exibido uma mensagem de confirmação. Isso apaga todo o conteúdo do disco de forma permanente, para prosseguir, selecionar a opção *YES*.

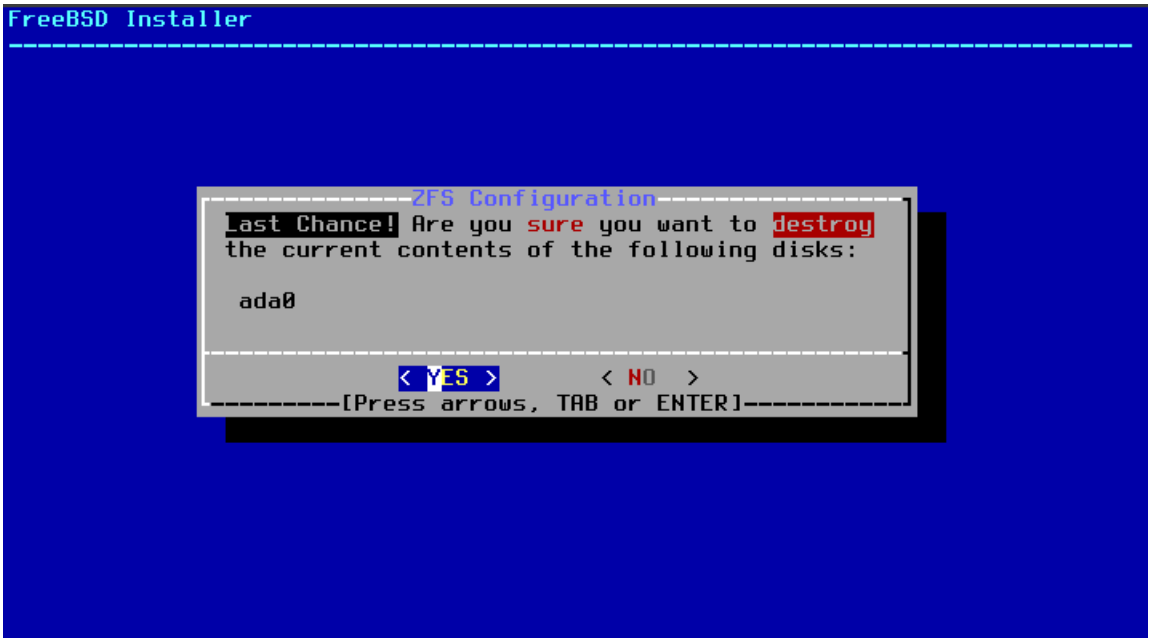


Figura 3.12: *Instalação OPNsense - Passo 9.*

Passo 10: Selecionar a opção *Complete Install* e pressionar o botão *OK*. Em seguida, o sistema será reinicializado com a instalação devidamente efetuada.

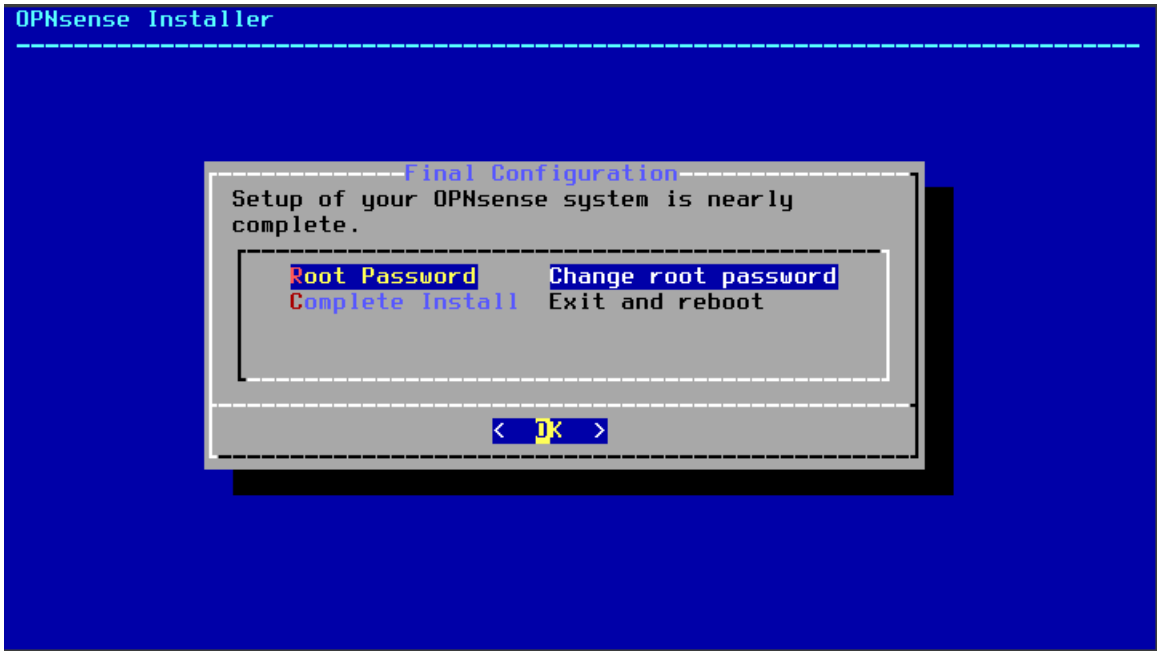


Figura 3.13: *Instalação OPNsense - Passo 10.*

3.4 Configuração do ambiente de HA

A configuração do ambiente de HA nos *firewalls* envolve o uso do protocolo CARP para compartilhar os endereços IP virtuais entre os dispositivos. Assim, se um dos *firewalls* falhar, o outro assume automaticamente suas funções e, também, um mecanismo que possa fazer a replicação das configurações entre eles.

3.4.1 Requisitos

Para configurar o ambiente de HA, os requisitos a seguir devem ser atendidos:

1. Os dois dispositivos OPNsense devem estar com a mesma versão de *software* instalada e um deles deve ser escolhido como principal;
2. Endereços IP dedicados:
   - Um endereço IP específico para cada interface presente no *firewall* (WANs, LANs);
   - Um endereço IP virtual compartilhado por interface com o CARP.
3. Uma interface exclusiva para a configuração do sincronismo dos estados e configurações dos *firewalls* (uma rede privada deve ser configurada nessas interfaces);
4. Os recursos XMLRPC *Sync* e pfSync devem ser ativados para que as configurações e os estados das conexões entre os dispositivos de *firewall* sejam sincronizados;
5. Os dois *firewalls* devem estar na mesma rede física ou lógica.

3.4.2 Configuração das interfaces de rede

A configuração do endereçamento IP presente nessa sessão utilizou endereços fictícios por motivos de segurança e sigilo da instituição.

Os *firewalls*, principal (*master*) e secundário (*backup*), foram nomeados respectivamente como OPN-FW-01 e OPN-FW-02 e suas interfaces de rede foram configuradas conforme a Tabela 3.3:

Tabela 3.3: *Tabela de Endereçamento dos Firewalls.*

Interfaces	OPN-FW-01	OPN-FW-02
WAN	192.168.254.11/24	192.168.254.12/24
HA	172.31.255.1/30	172.31.255.2/30
LAGG-IFPB	SEM ENDEREÇO	SEM ENDEREÇO
DMZ-ADM	192.168.101.1/24	192.168.101.2/24
DMZ-ACAD	192.168.102.1/24	192.168.102.2/24
LAN-ADM	192.168.111.1/24	192.168.111.2/24
LAN-ACAD	192.168.112.1/24	192.168.112.2/24

A interface LAGG não possui endereço IP, pois seu papel é agregar as interfaces físicas em um *link* lógico, que é usado para transportar o tráfego através das VLANs até o *switch core*. Os endereços configurados no OPNsense podem ser observados nas Figuras 3.14 e 3.15:

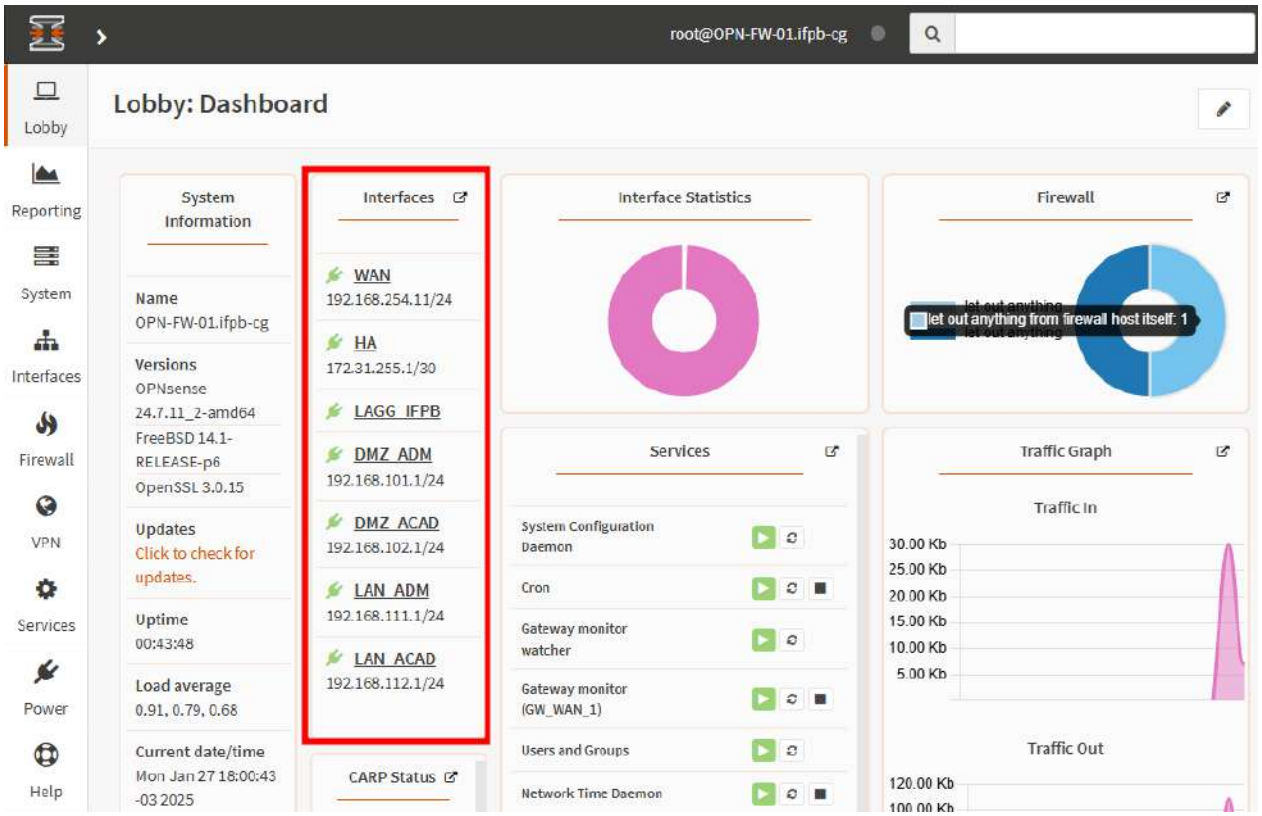


Figura 3.14: *Endereços IP OPN-FW-01.*

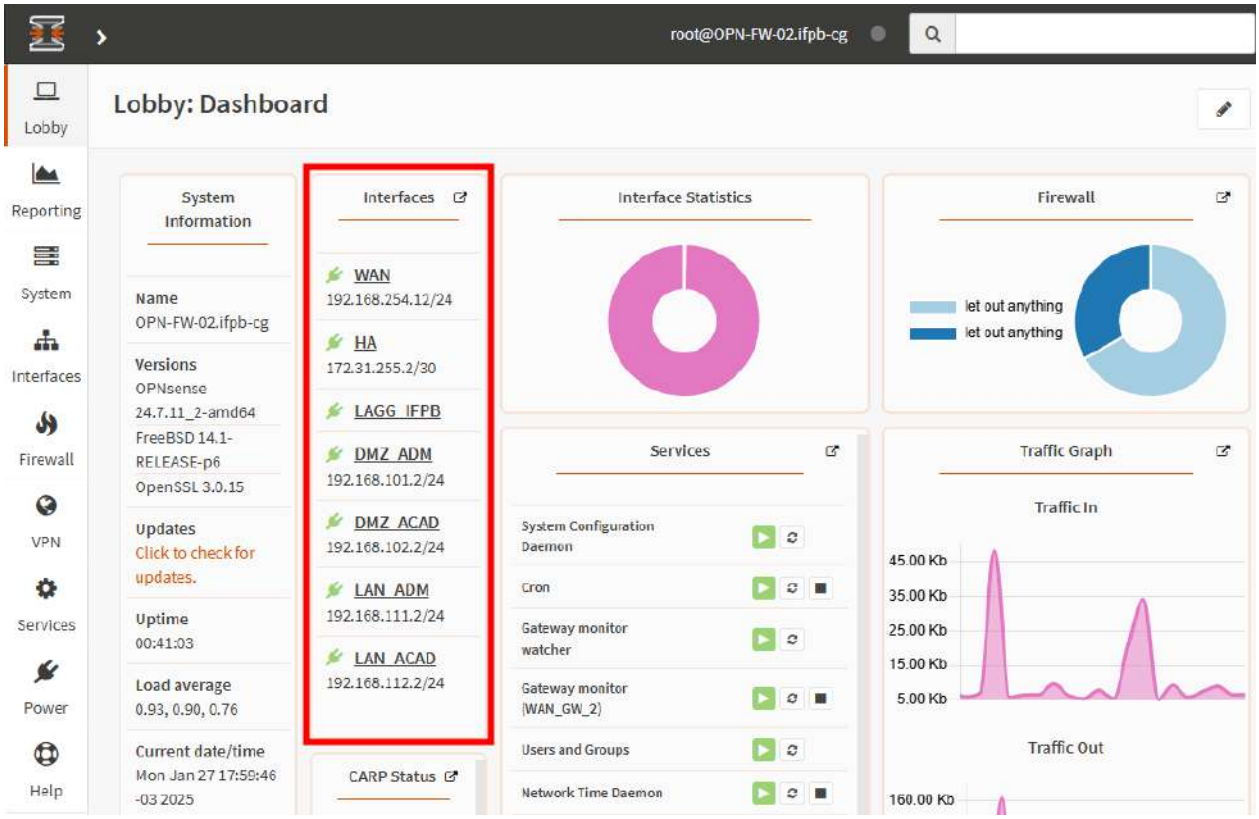


Figura 3.15: *Endereços IP OPN-FW-02.*

3.4.3 Configuração do CARP

Para inciar a configuração do CARP é preciso definir os endereços IP virtuais (VIP) que estão associados aos *virtual host ID* (vhid) como já observados na Figura 2.1.

Os VIPs foram definidos em ambos os *firewalls* conforme a Tabela 3.4:

Tabela 3.4: *Configuração dos endereços VIP.*

Mode	CARP	CARP	CARP	CARP	CARP
Interface	WAN	DMZ-ADM	DMZ-ACAD	LAN-ADM	LAN-ACAD
Network/ Address	192.168.254 .5/24	192.168.101 .254/24	192.168.102 .254/24	192.168.111 .254/24	192.168.112 .254/24
Password	opnsense	opnsense	opnsense	opnsense	opnsense
VHID Group	1	2	3	4	5
Description	VIP-WAN	VIP-DMZ-ADM	VIP-DMZ-ACAD	VIP-LAN-ADM	VIP-LAN-ACAD

Abaixo, as informações da Tabela 3.4:

1. ***Mode*:** Refere-se ao tipo de endereço IP virtual utilizado, o CARP é usado para ativar o HA;
2. ***Interface*:** É a interface que responderá pelo endereço IP virtual;
3. ***Network/Address*:** É o próprio endereço IP virtual seguido pelo comprimento do prefixo de rede;
4. ***Password*:** É obrigatório a utilização de uma senha para assegurar o processo de comunicação entre os membros do HA, foi utilizado `opnsense` para exemplo didático;
5. **VHID *Group*:** É o grupo virtual de identificação do *host*, onde cada VIP do CARP deve ter o seu identificador único, porém, deve ser o mesmo para os mesmos VIPs em ambos os *firewalls*;
6. ***Description*:** Descrição para facilitar a identificação do VIP.

Como é possível observar, nas Figuras 3.16 a 3.20, a configuração dos VIPs deve ser aplicada em ambos os *firewalls*:

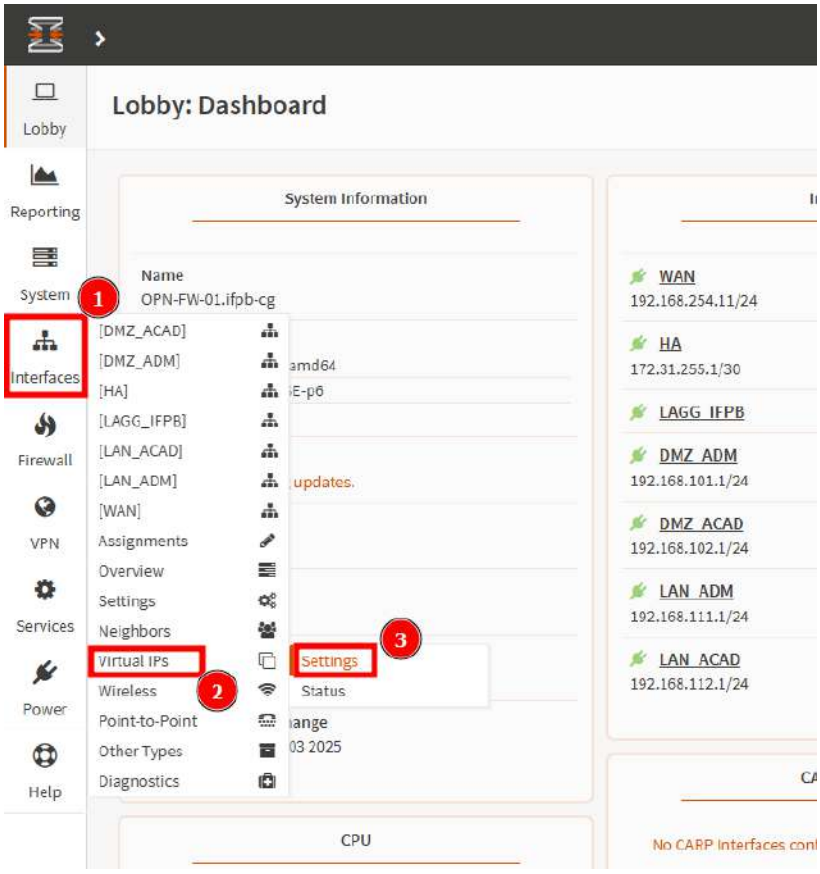


Figura 3.16: *Configuração dos VIPs*

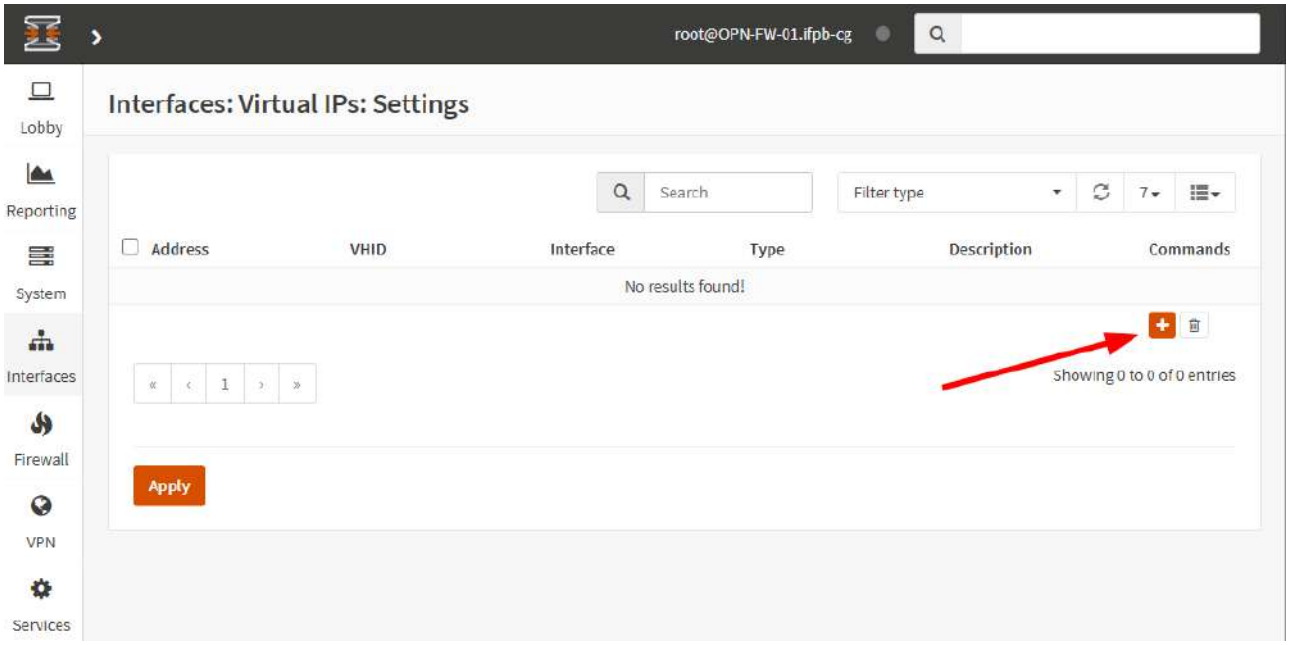


Figura 3.17: *Adição dos VIPs*

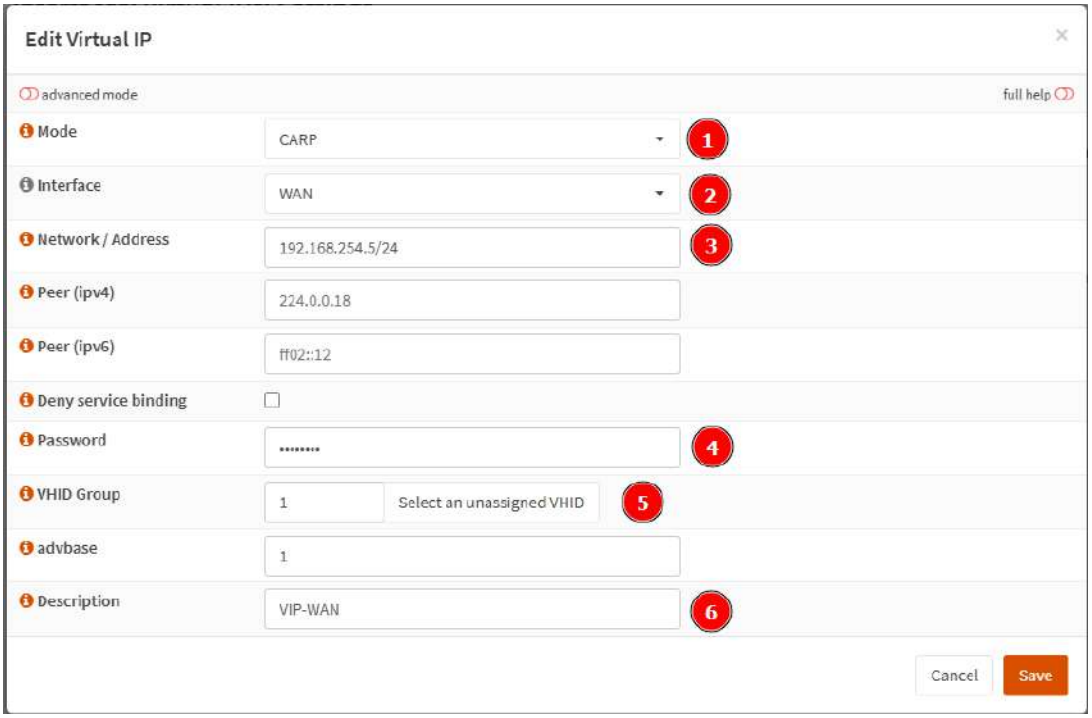


Figura 3.18: *Configuração do VIP da interface WAN*

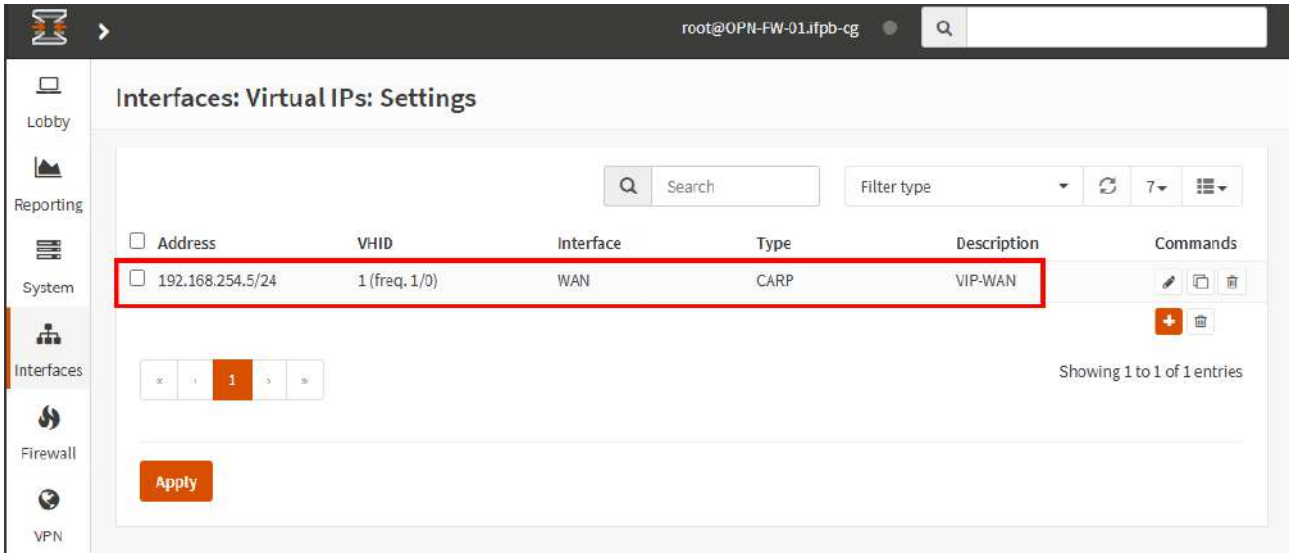


Figura 3.19: *VIP da interface WAN configurada*

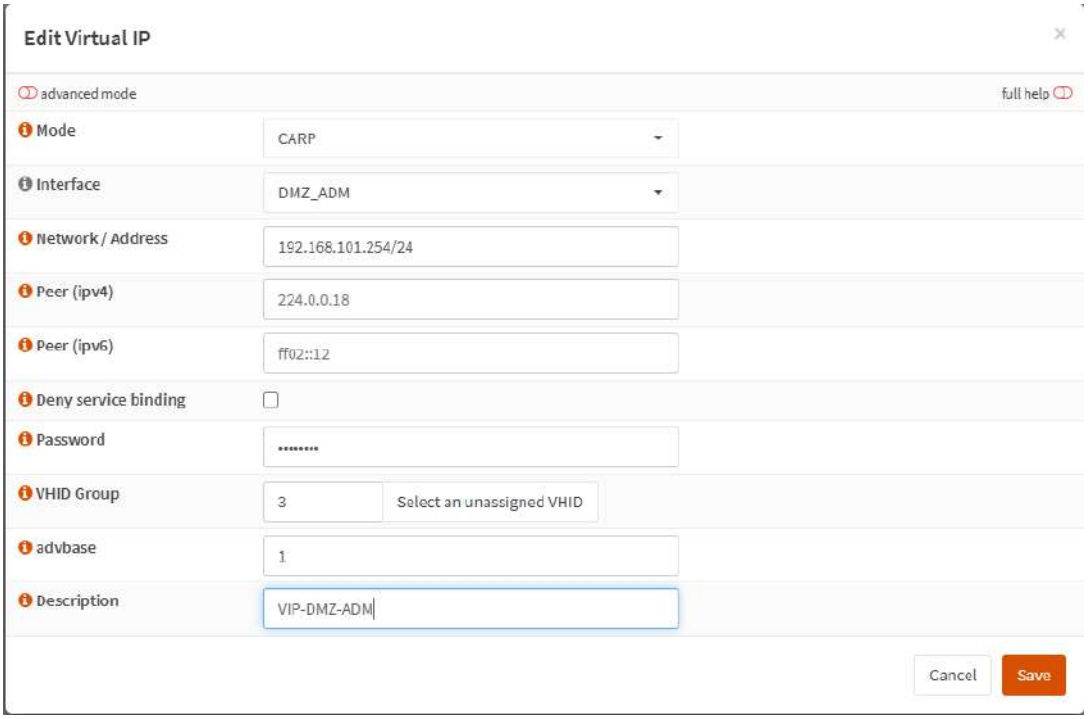


Figura 3.20: *Configuração dos VIP para uma das interfaces LAN*

A seguir, as Figuras 3.21 e 3.22 ilustram as configurações dos VIPs em ambos os *firewalls*:

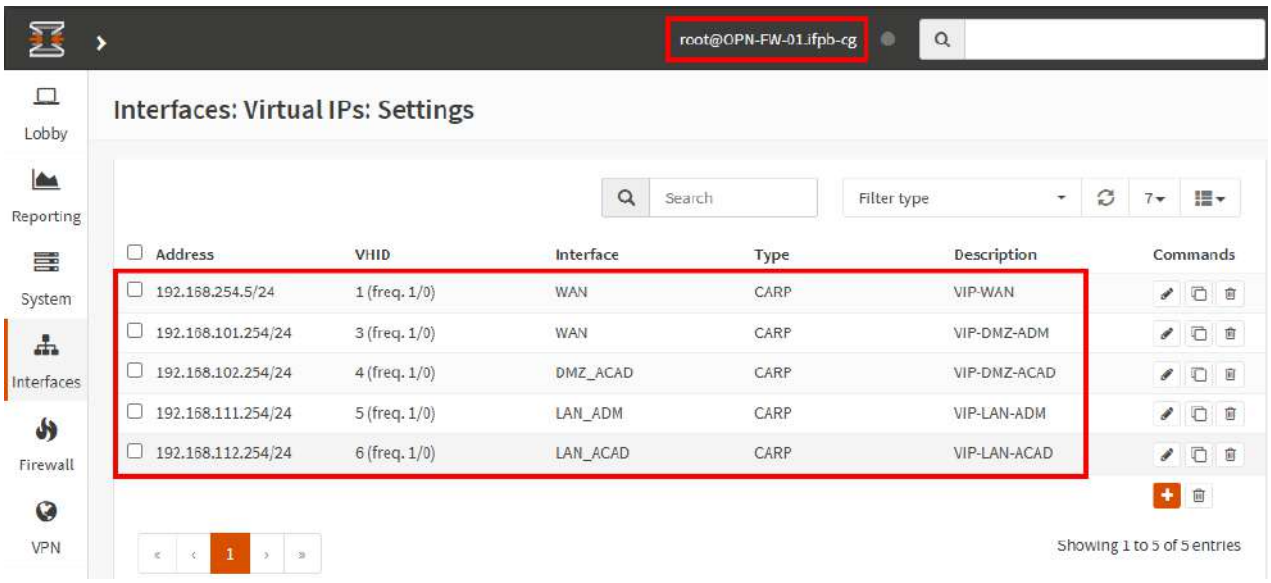


Figura 3.21: *Configuração dos VIP no OPN-FW-01*

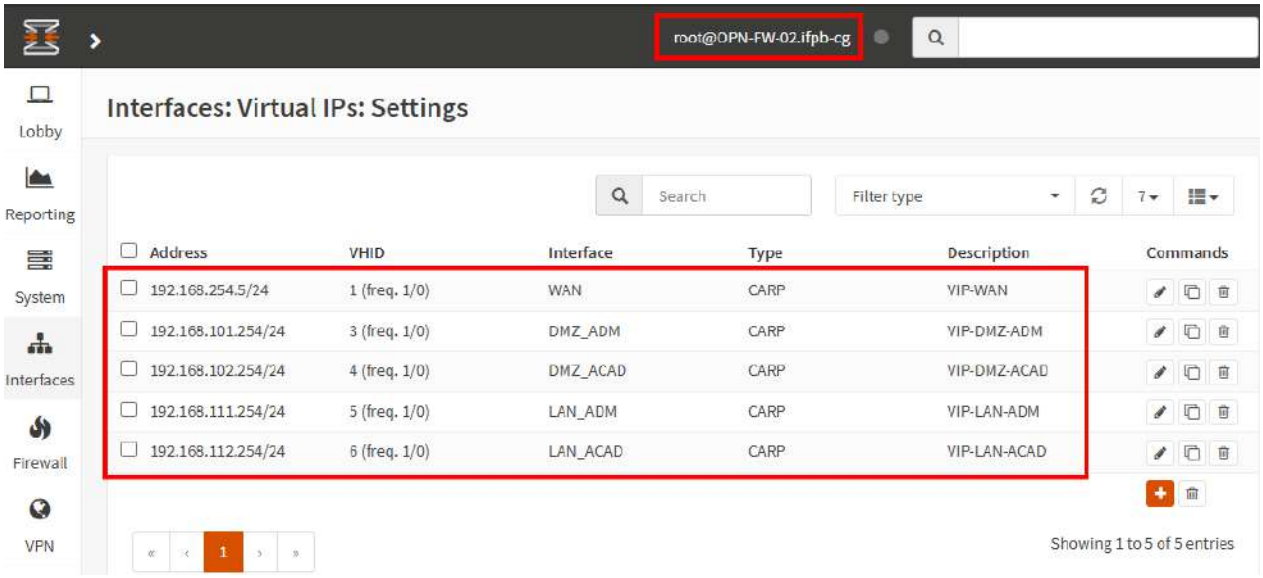


Figura 3.22: *Configuração dos VIP no OPN-FW-02*

3.4.4 Configuração do NAT

Configurar o NAT é essencial para funcionamento do ambiente, para isso, o primeiro passo é mudar o modo de operação padrão do NAT de saída (*outbound)* no OPNsense de automático, para o modo híbrido, conforme as Figuras 3.23 e 3.24 a seguir:

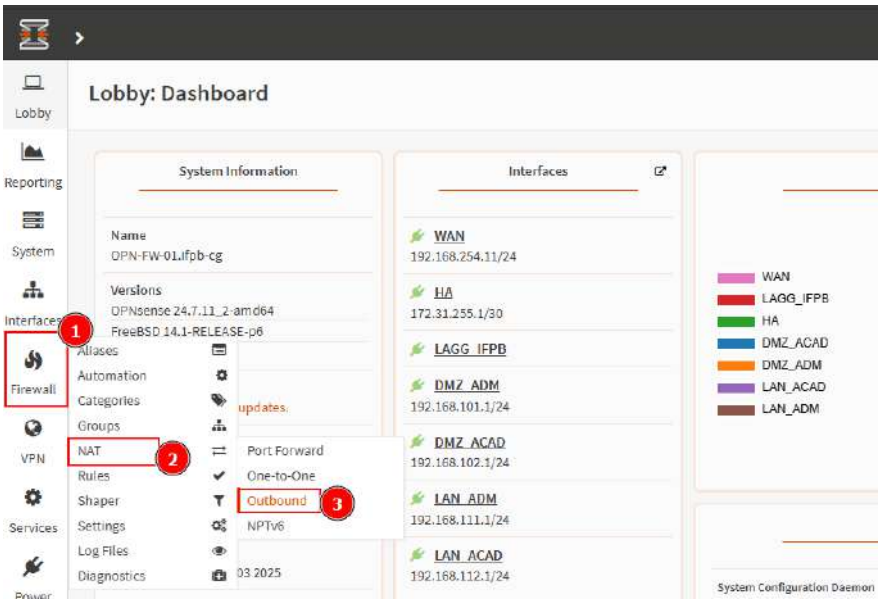


Figura 3.23: *Acesso ao menu de configuração do NAT Outbound*

No painel de controle na lateral esquerda: *Firewall* →NAT →*Outbound*

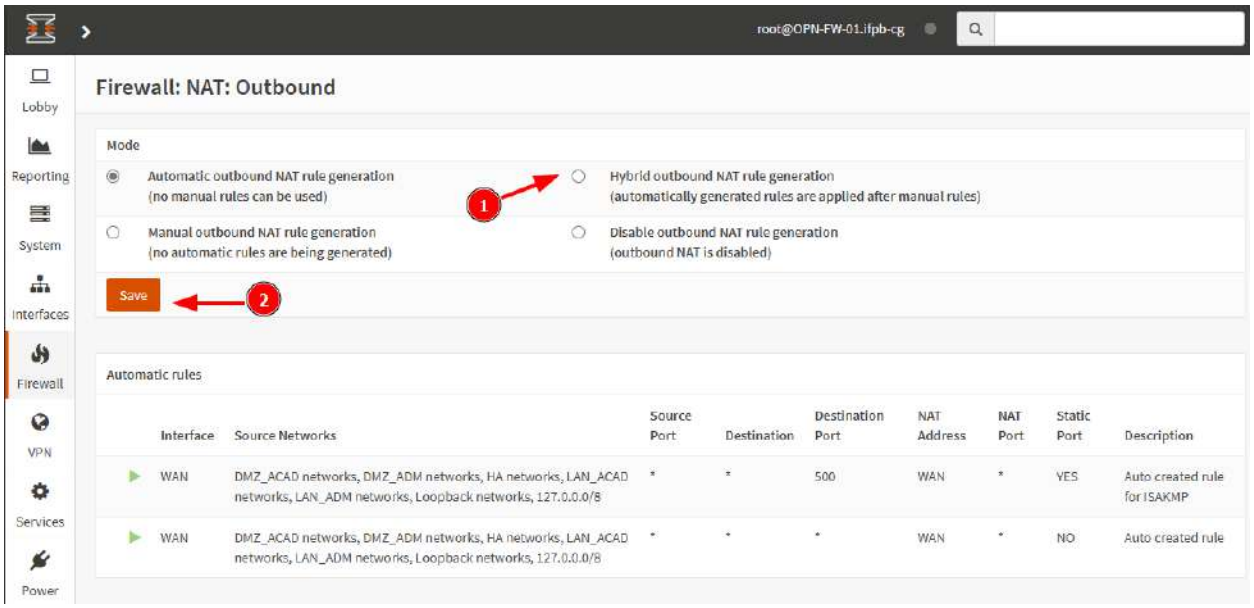


Figura 3.24: *Alteração do modo de NAT Outbound*

A alteração do modo de operação do NAT de saída é necessária, para que o tráfego da rede interna possa ser encaminhado para a internet, e também, através da interface de HA. As regras foram criadas em ambos os *firewalls*, e para cada rede presente na configuração do CARP, mas, para facilitar a operação e minimizar a quantidade de regras necessárias, foi utilizado o recurso *aliases*. Com ele, é possível criar um grupo com todas as redes de origem que foram utilizadas nas regras do NAT, como mostram as Figuras de 3.25 a 3.28:

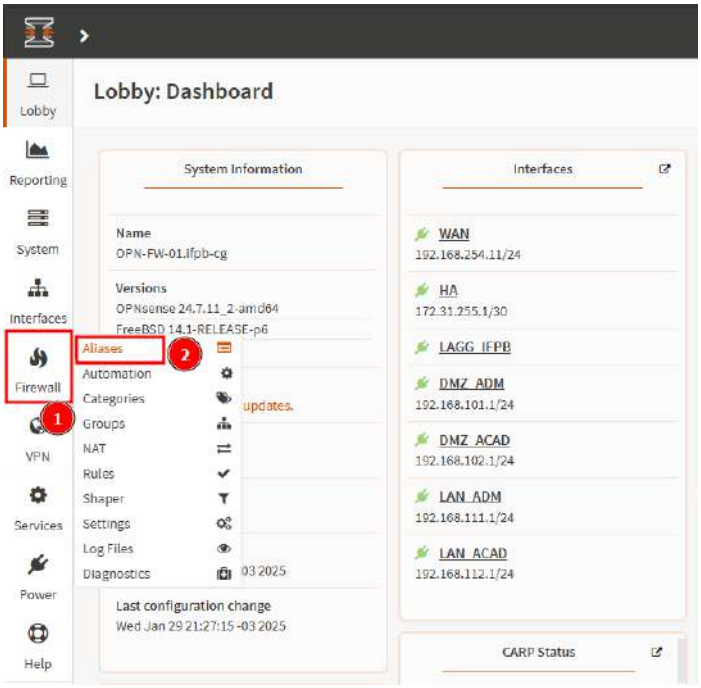


Figura 3.25: *Criação de alias para redes de origem.*

No painel de controle na lateral esquerda: *Firewall* →*Aliases*.

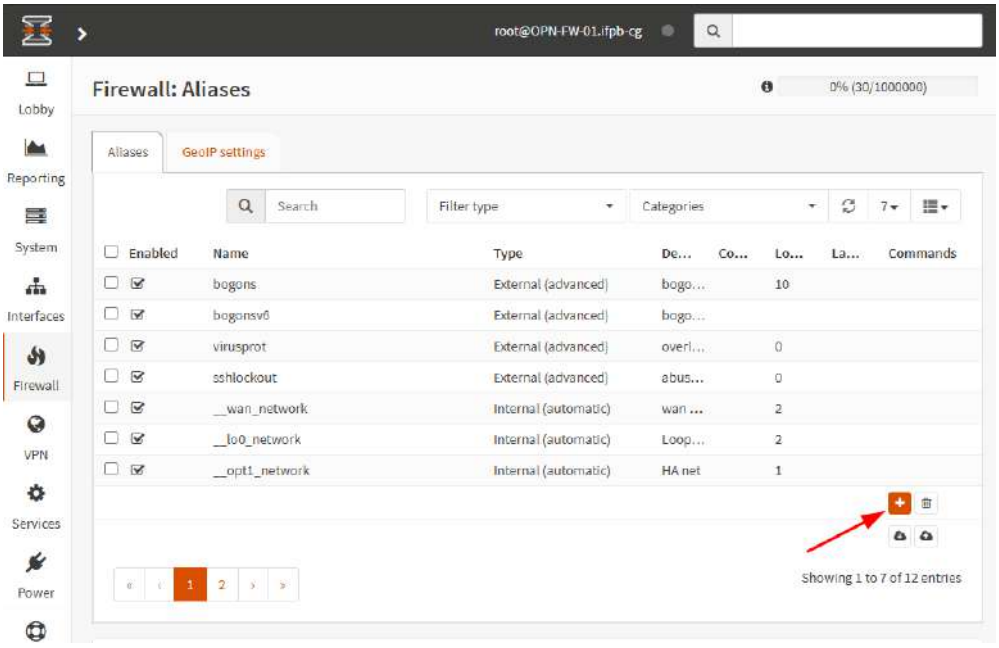


Figura 3.26: *Adição dos aliases.*

Nesse ponto, adicionar um *alias* para agrupar os endereços das redes criadas nos *firewalls*, tornam a criação das regras mais fácil e elegante.

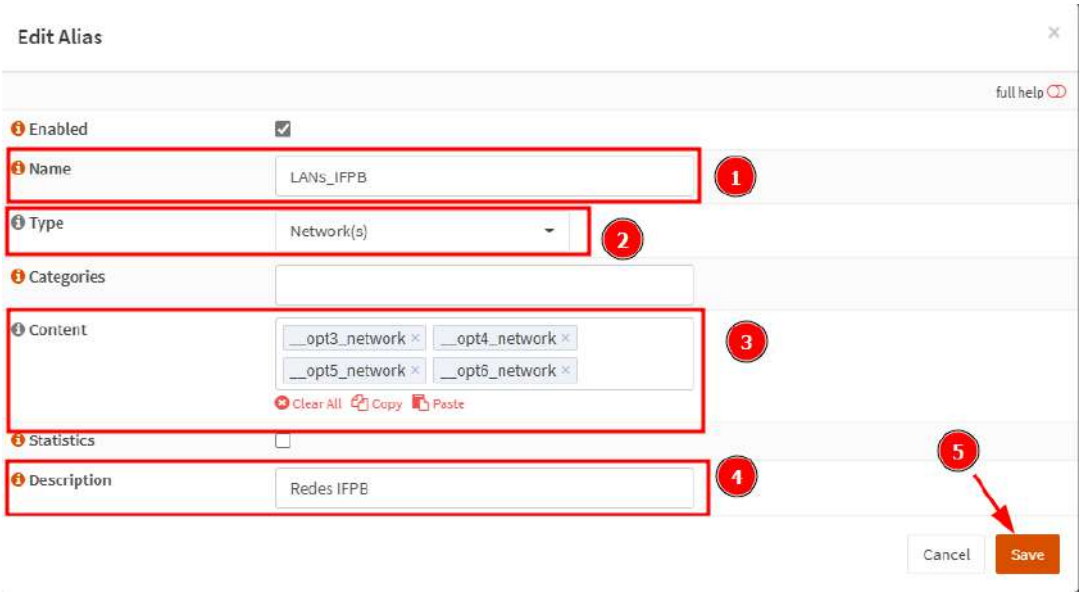


Figura 3.27: *Definição das informações do alias.*

Para configurar um novo *alias*:

1. ***Name*:** Definir um nome para o *alias*;
2. ***Type*:** Selecionar tipo como *Network(s)*;
3. ***Content*:** Selecionar as redes de origem criadas e adicionadas anteriormente ao CARP;
4. ***Description*:** Definir uma descrição que facilite a identificação do *alias* pelo administrador.

Preenchidos os campos necessários, basta salvar as configurações, assim o *alias* estará cadastrado (Figura 3.28):

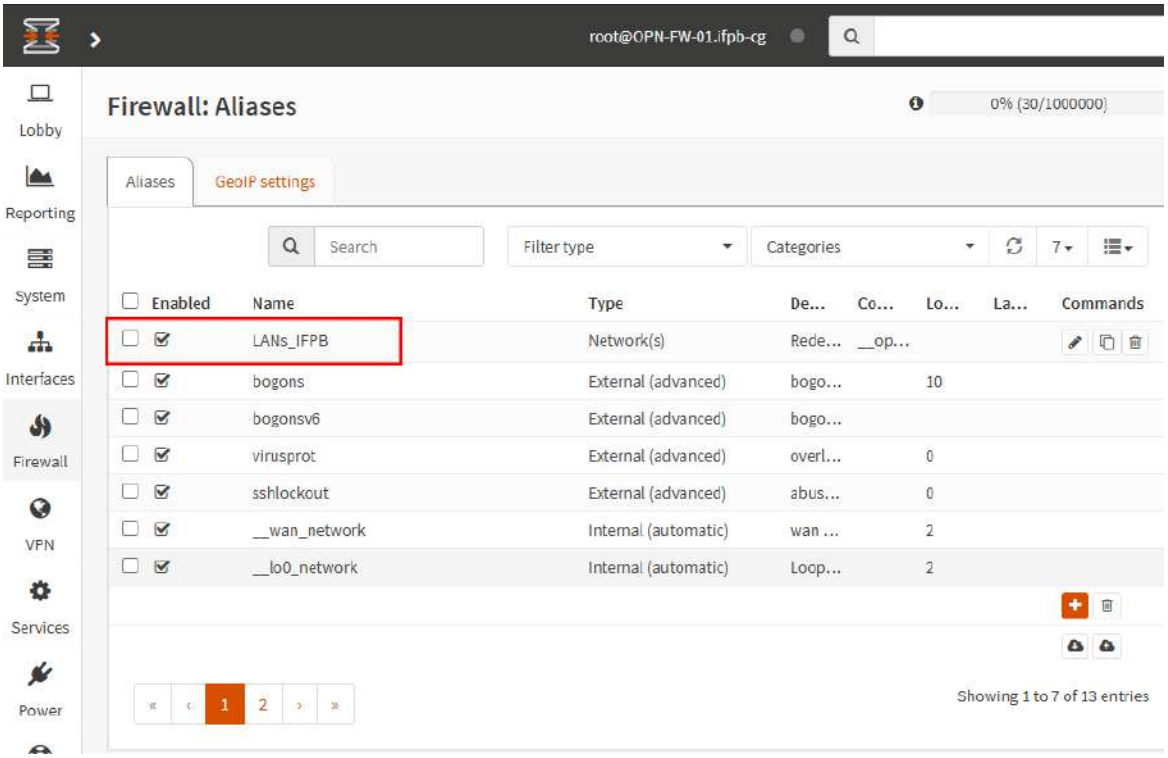


Figura 3.28: *Visualização do alias criado.*

Com *aliases* criados, continuar com a configuração do NAT e criar as regras necessárias. No painel de controle na lateral esquerda, selecionar novamente: *Firewall* →NAT →*Outbound* e em seguida, adicionar uma nova regra conforme Figuras 3.29 e 3.30.

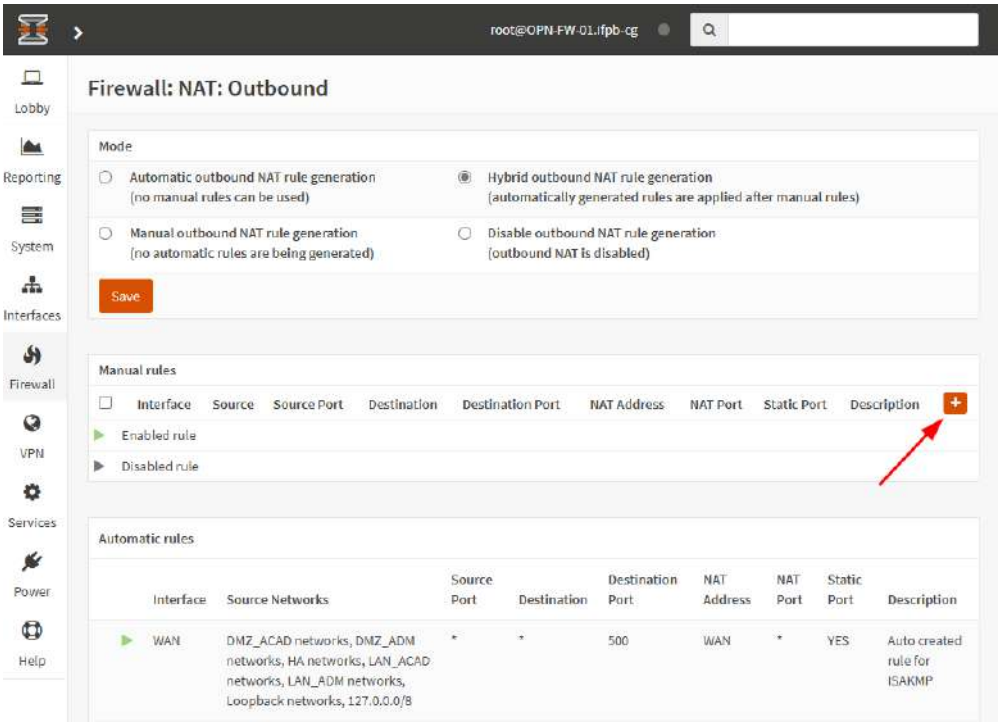


Figura 3.29: *Adição da regra de NAT.*

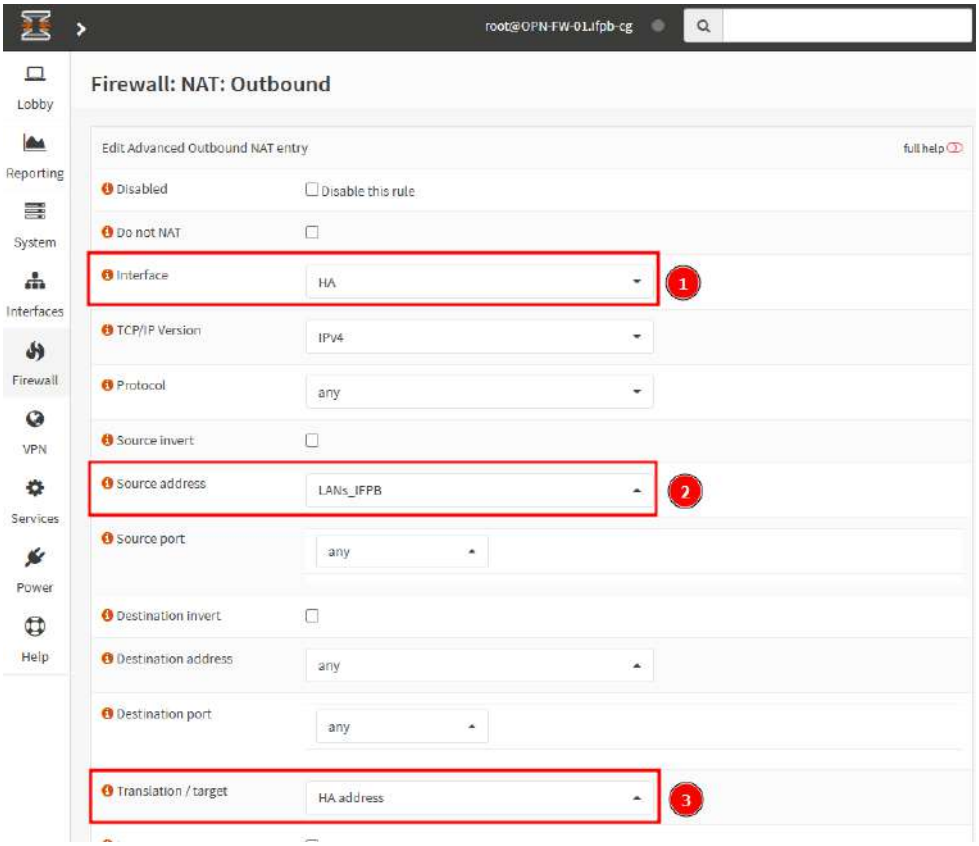


Figura 3.30: *Configuração da regra de NAT através da interface de HA.*

Na criação da nova regra, modificar as seguintes informações:

1. ***Interface*:** Selecionar a interface do HA;
2. ***Source address*:** Selecionar o *alias* criado anteriormente com as redes de origem selecionadas;
3. ***Translate/target*:** Selecionar o endereço da interface HA, que será utilizado na tradução do NAT;
4. Salvar a regra.

Agora, repetir o processo para adicionar a regra seguinte, conforme Figura 3.31.

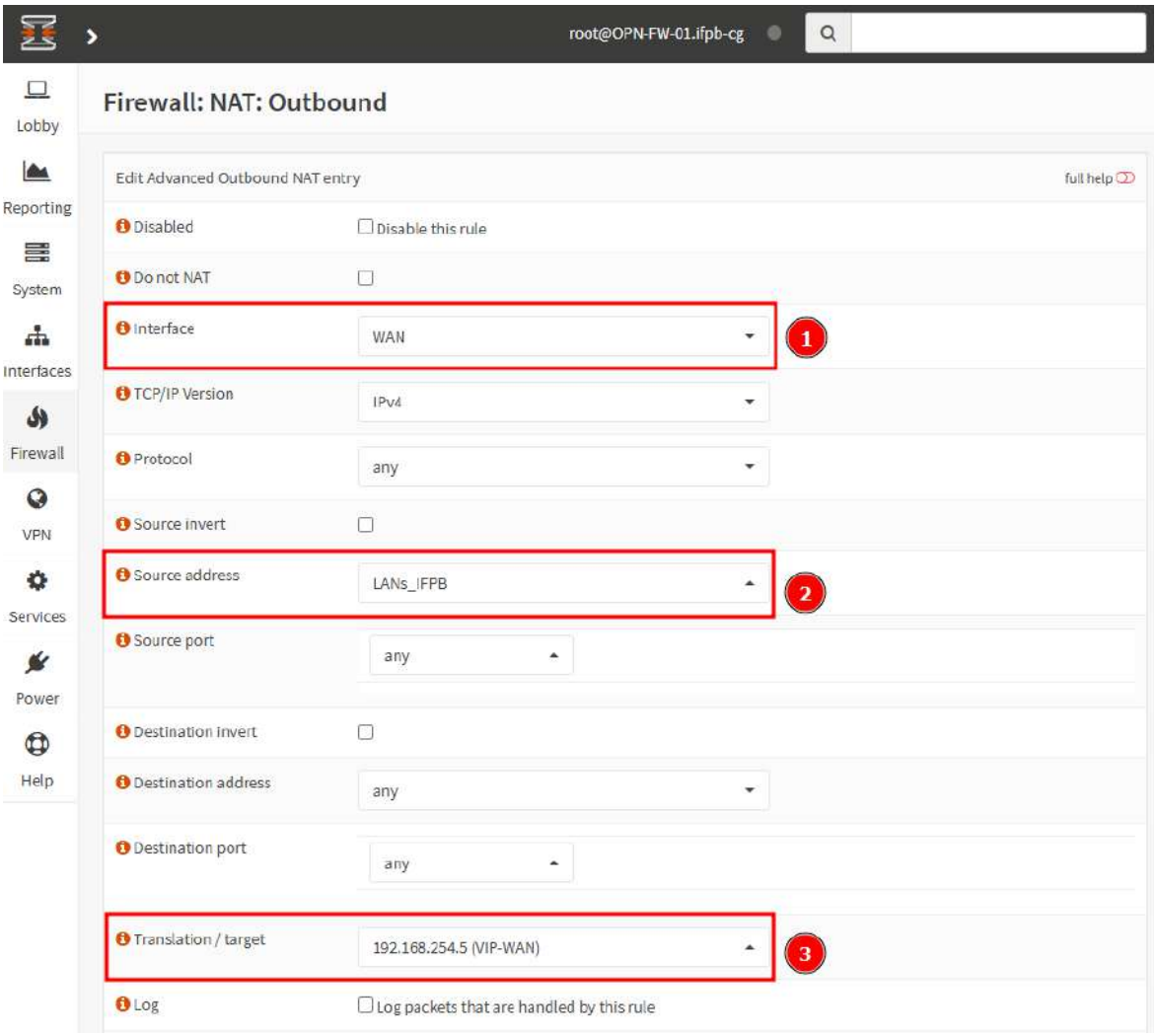


Figura 3.31: *Configuração da regra de NAT através da interface WAN.*

1. ***Interface*:** Selecionar a interface do WAN;
2. ***Source address*:** Selecionar o *alias* criado anteriormente com as redes de origem selecionadas;
3. ***Translate/target*:** Selecionar o endereço VIP-WAN, que será utilizado na tradução do NAT;
4. Salvar a regra e aplicar as alterações na próxima tela.

As regras criadas são vistas a seguir (Figura 3.32):

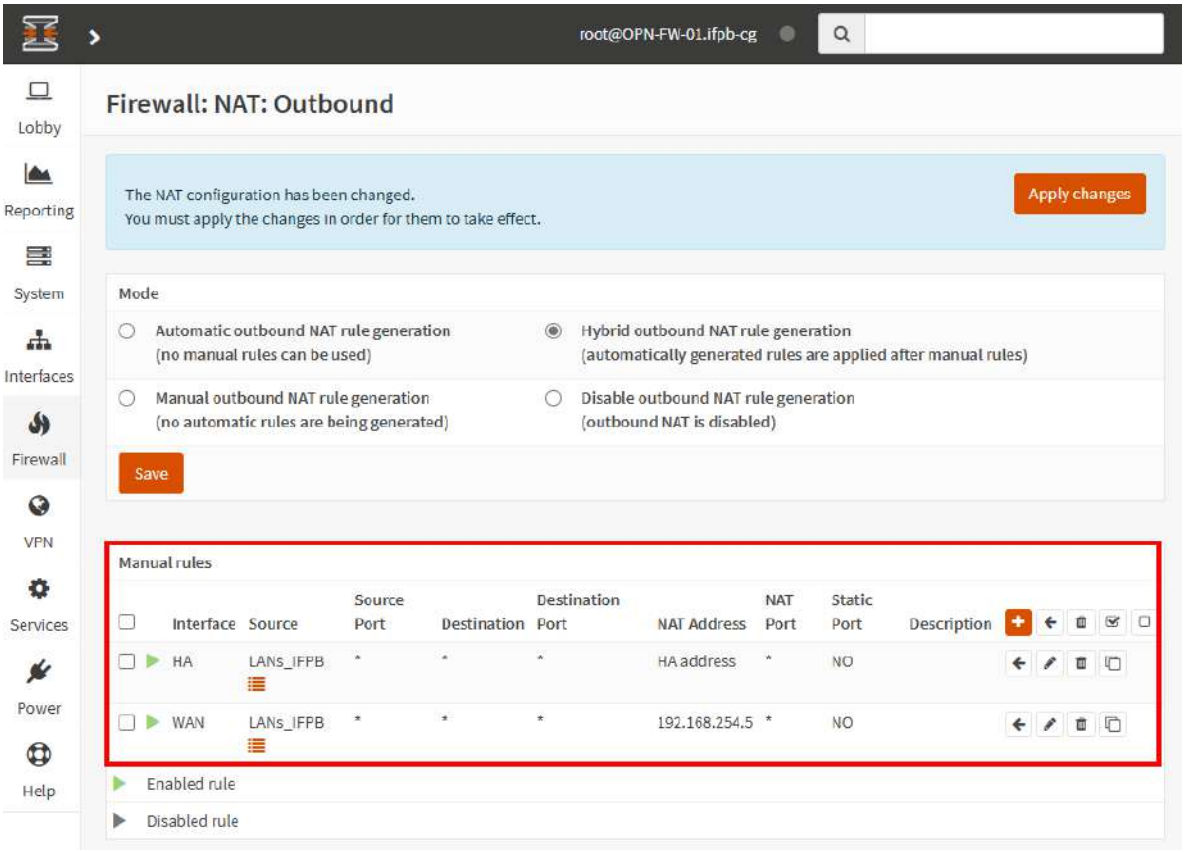


Figura 3.32: *Visualização das regras de NAT criadas.*

3.4.5 Ativação do HA

Depois de realizadas as configurações preliminares necessárias, é exibido nas Figuras 3.33 e 3.34, como ativar o HA, primeiro no *firewall master*:

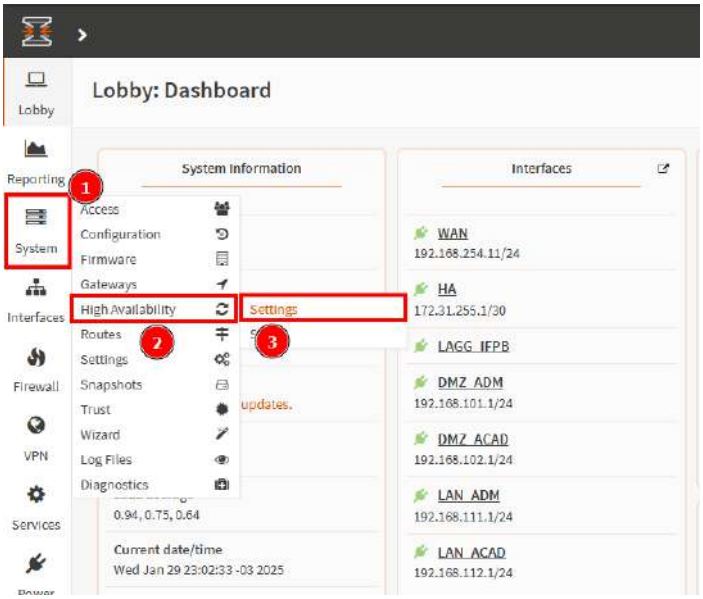


Figura 3.33: *Ativação o HA no firewall master.*

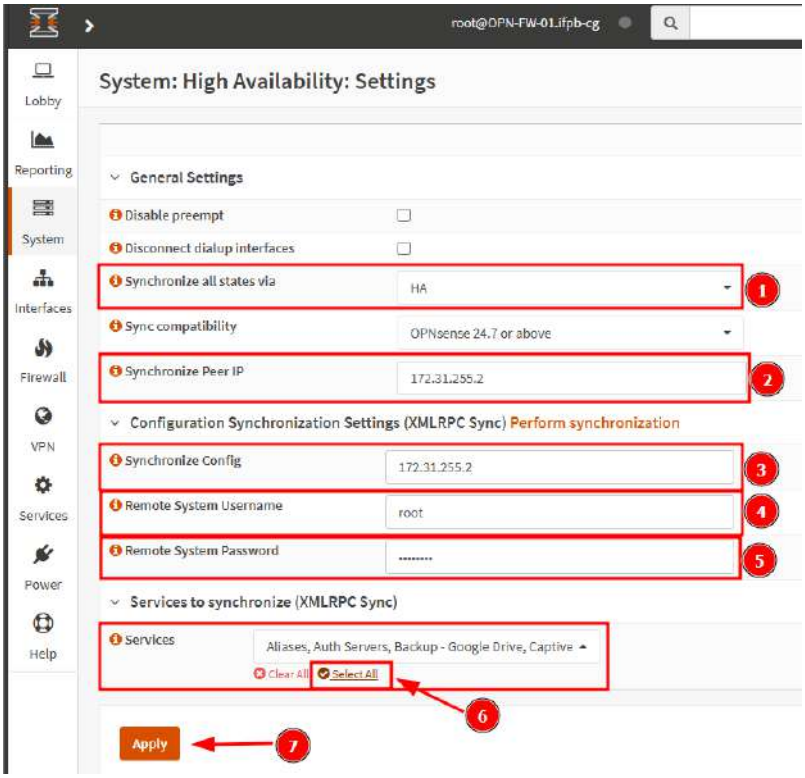


Figura 3.34: *Configuração dos parâmetros do HA no firewall master.*

1. ***Synchronize all states via*:** Selecionar a interface que foi configurada para o HA para a sincronização dos estados das conexões;

2. ***Synchronize Peer IP*:** Informar o endereço IP do *firewall backup*;

3. ***Synchronize Config*:** Informar o endereço IP do *firewall backup* para a sincronização das configurações;

4. ***Remote System Username*:** Informar o usuário que será usado no processo de sincronização das configurações para o *firewall backup*;

5. ***Remote System Password*:** Informar a senha do usuário;

6. ***Services*:** Informar quais serviços terão suas configurações sincronizadas com o *firewall backup*;

7. Aplicar as configurações.

As configurações para o *firewall backup* são mais simples, como mostrado na Figura 3.35

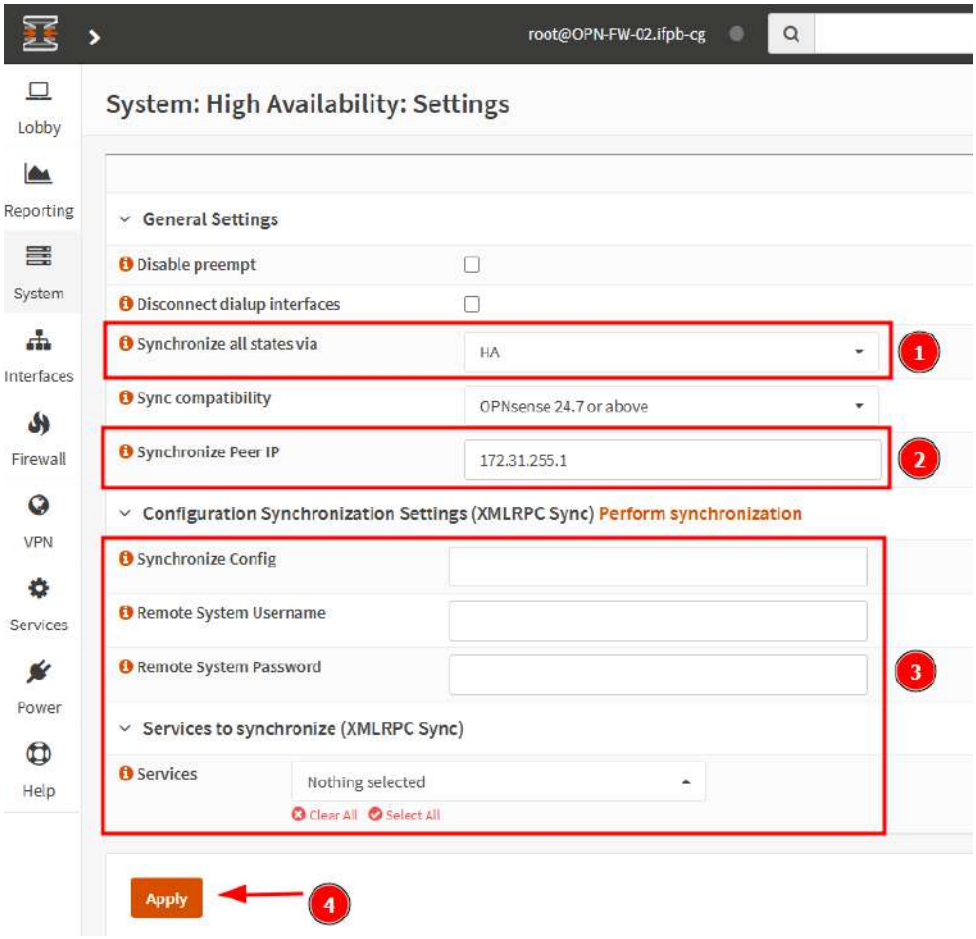


Figura 3.35: *Ativação do HA no firewall backup.*

É preciso selecionar apenas a interface de HA para a sincronização dos estados das conexões e definir o endereço IP do *master*, pois, *backup* não irá sincronizar nenhuma configuração com o *master*, e em seguida, aplicar as configurações.

Nas Figuras 3.36 e 3.37, observa-se o *status* depois da configuração no *master* e *backup*:

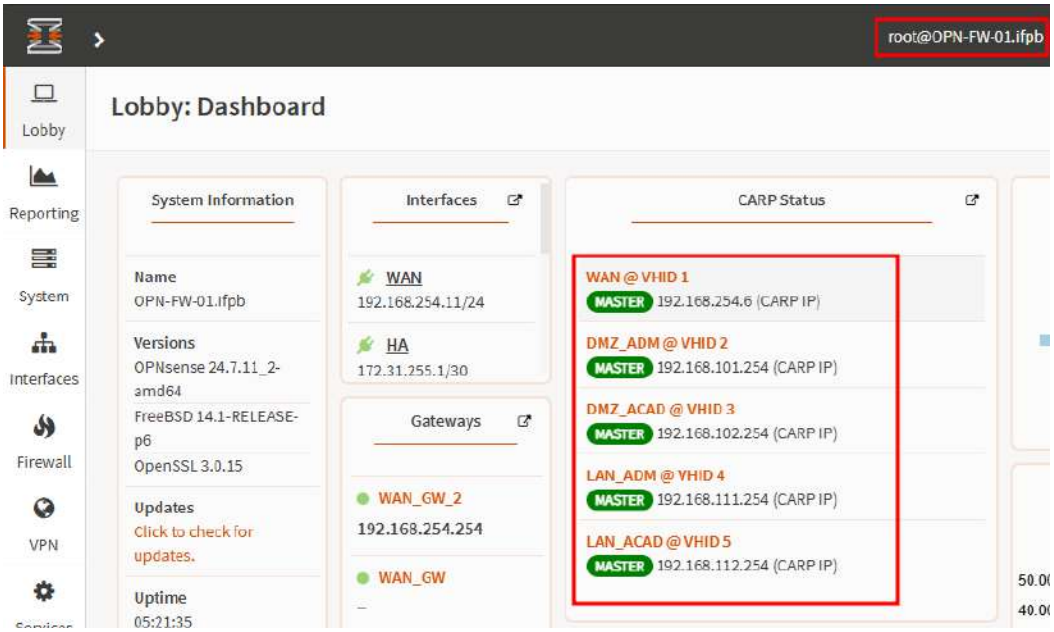


Figura 3.36: *Verificação do status do HA no firewall master.*

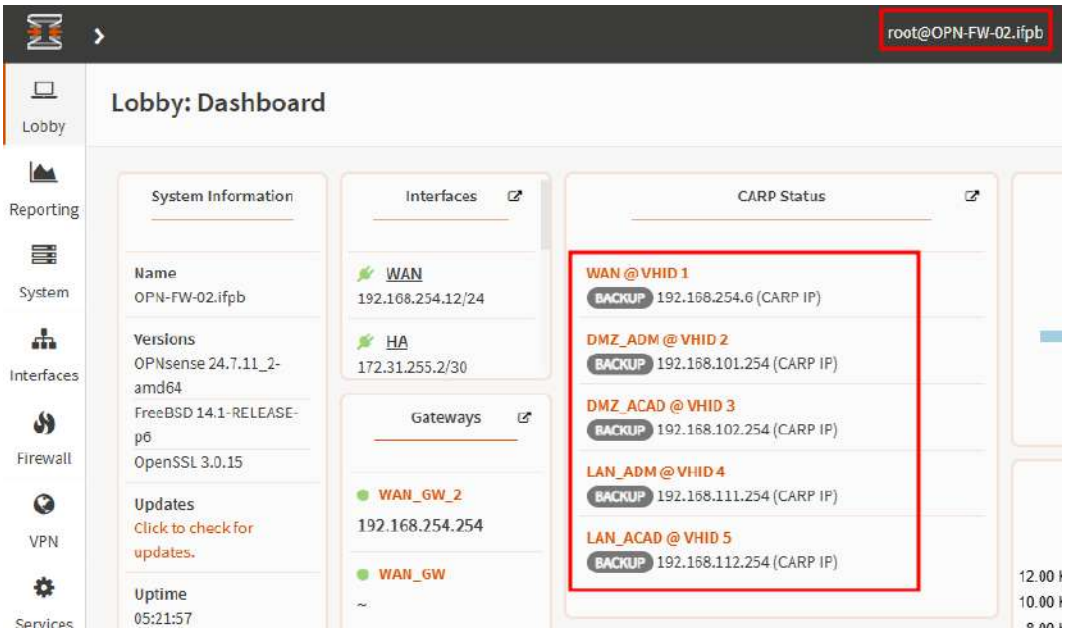


Figura 3.37: *Verficação do status do HA no firewall backup.*

3.4.6 Testes do ambiente de HA

Foram utilizadas as ferramentas ping[2] e gping[3] para a validação das configurações do ambiente de HA. Com o ping, foi possível aferir a latência nas comunicações e a perda de

[2]O ping é um utilitário que usa o protocolo ICMP para testar a conectividade entre equipamentos.

[3]GPing ou *Graphical* ping, é uma ferramenta que utiliza os resultados obtidos pelo ping para criar gráficos em tempo real.

pacotes no momento em que as falhas foram simuladas, já com o gping, além das informações geradas pelo ping, foi possível também visualizar o comportamento do tráfego de teste em formato de gráfico

3.5 Migração das regras e políticas de *firewall*

Como o iptables é uma plataforma diferente do OPNsense, suas regras foram migradas de forma manual depois de analisadas. O primeiro passo foi mapear as regras do iptables para o OPNsense, seguindo a estratégia abaixo:

Chains

O mapeamento entre as *chains* da tabela *filter* do iptables para a estrutura no OPNsense, como detalhado na Tabela 3.5:

Tabela 3.5: *Chains iptables.*

iptables	**OPNsense**
INPUT ou FORWARD	*Firewall* →*Rules* →Interface (WAN/LAN etc)

NAT

Na Tabela 3.6, é detalhado o mapeamento entre as *chains* da tabela NAT do iptables para a sessão de NAT do OPNsense:

Tabela 3.6: *NAT iptables.*

iptables	**OPNsense**
PREROUTING (DNAT)	*Firewall* →NAT →*Port Forward*
POSTROUTING (SNAT/*Masquerade*)	*Firewall* →NAT →*Outbound*

A tarefa de mapear e reescrever as regras demanda um trabalho um pouco mais cuidadoso, pois, traduzir as regras de forma eficiente e correta é essencial para o funcionamento dos serviços de redes, assim como, as políticas de segurança, filtragem e tradução de endereços IP.

Capítulo 4

Resultados dos experimetos

Nesse tópico, os resultados simulando a interrupção dos serviços de *firewall* usando o modelo da topologia atual aplicada no *campus* e a topologia proposta no trabalho serão apresentados.

4.1 Resultados com a topologia atual do *campus*

Usando a topologia atual do *campus*, os testes de conectividade para os seguintes destinos foram realizados:

- **192.168.111.254**: Endereço IP do *firewall* diretamente conectado à rede de origem dos testes;
- **1.1.1.1**: Endereço IP público do serviço de DNS público da *Cloudflare*;
- **8.8.8.8**: Endereço IP público do serviço de DNS público do *Google*.

É possível observar nas Figuras 4.1 e 4.2 o comportamento do tráfego relacionado às respostas obtidas através das ferramentas Ping e GPing. Com o funcionamento correto dos serviços de *firewall*, o tráfego flui sem interrupções e sem instabilidades.

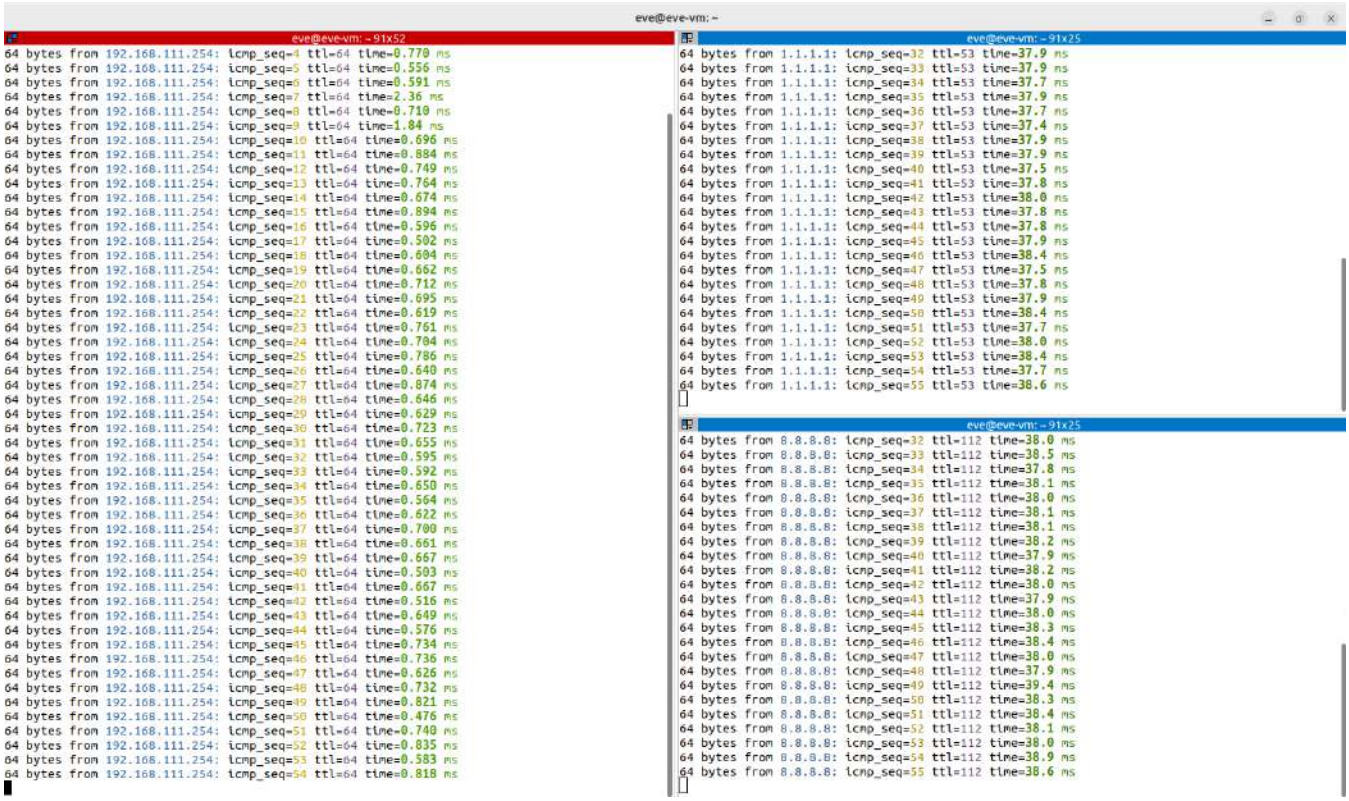


Figura 4.1: *Teste de ping para os destinos citados.*

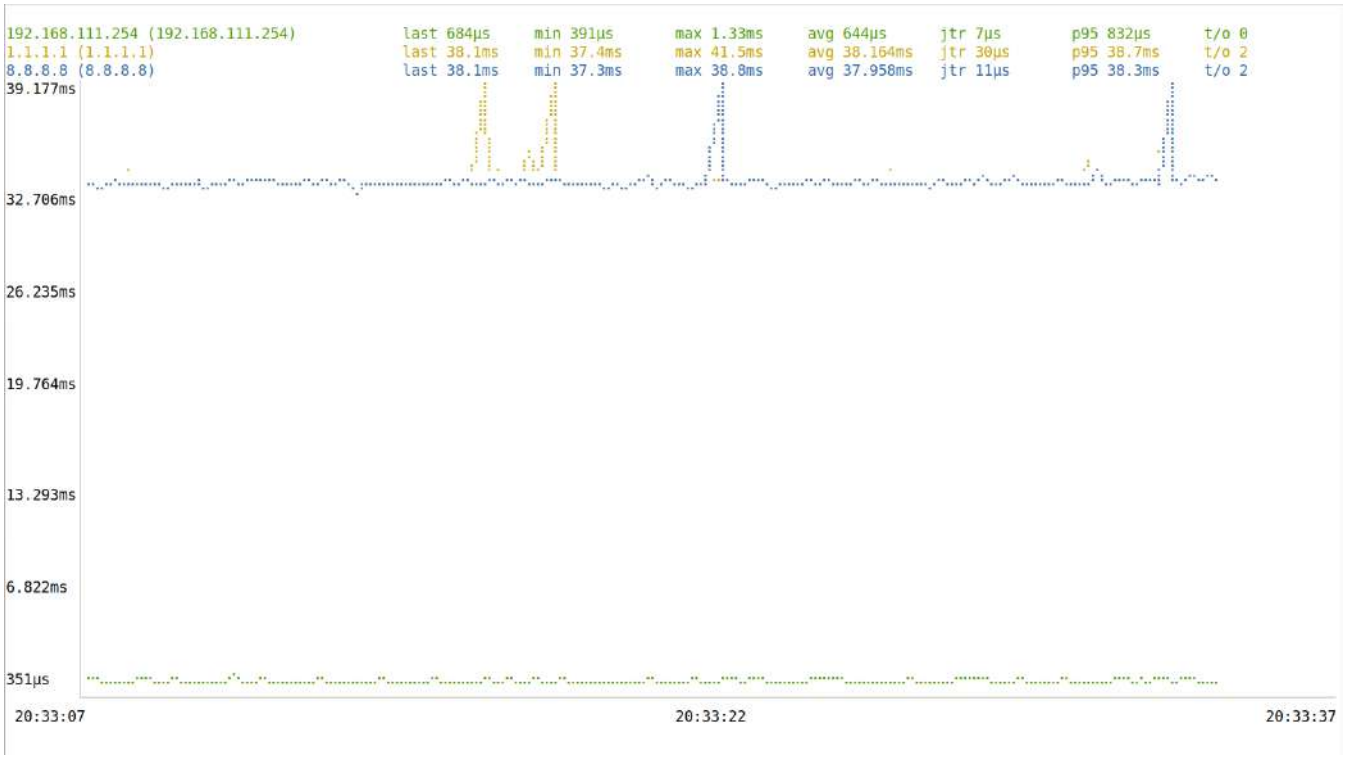


Figura 4.2: *Gráfico com o resultado do teste para os destinos citados.*

Conforme a Figura 4.3, foi simulado uma interrupção do *firewall*. No ambiente real, essa falha pode ocorrer por diversos motivos, como: problemas elétricos, falhas de *hardware*, *software*. o que implica em desligamento do sistema, ou até , falha nas interfaces de rede que interligam as redes internas.

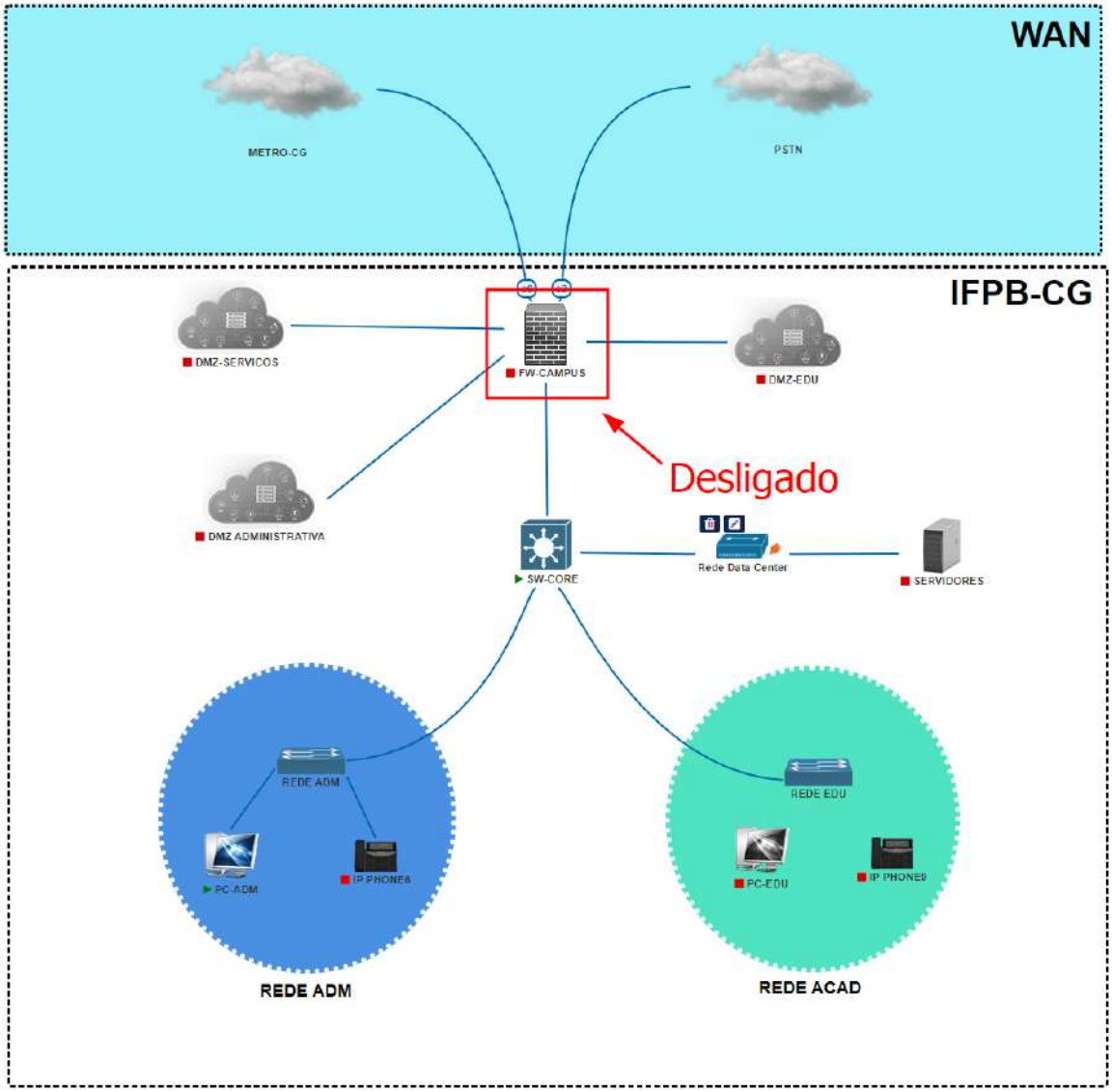


Figura 4.3: *Firewall desligado por falha.*

As Figuras 4.4 e 4.5, ilustram o comportamento do tráfego no momento da falha do *firewall*:

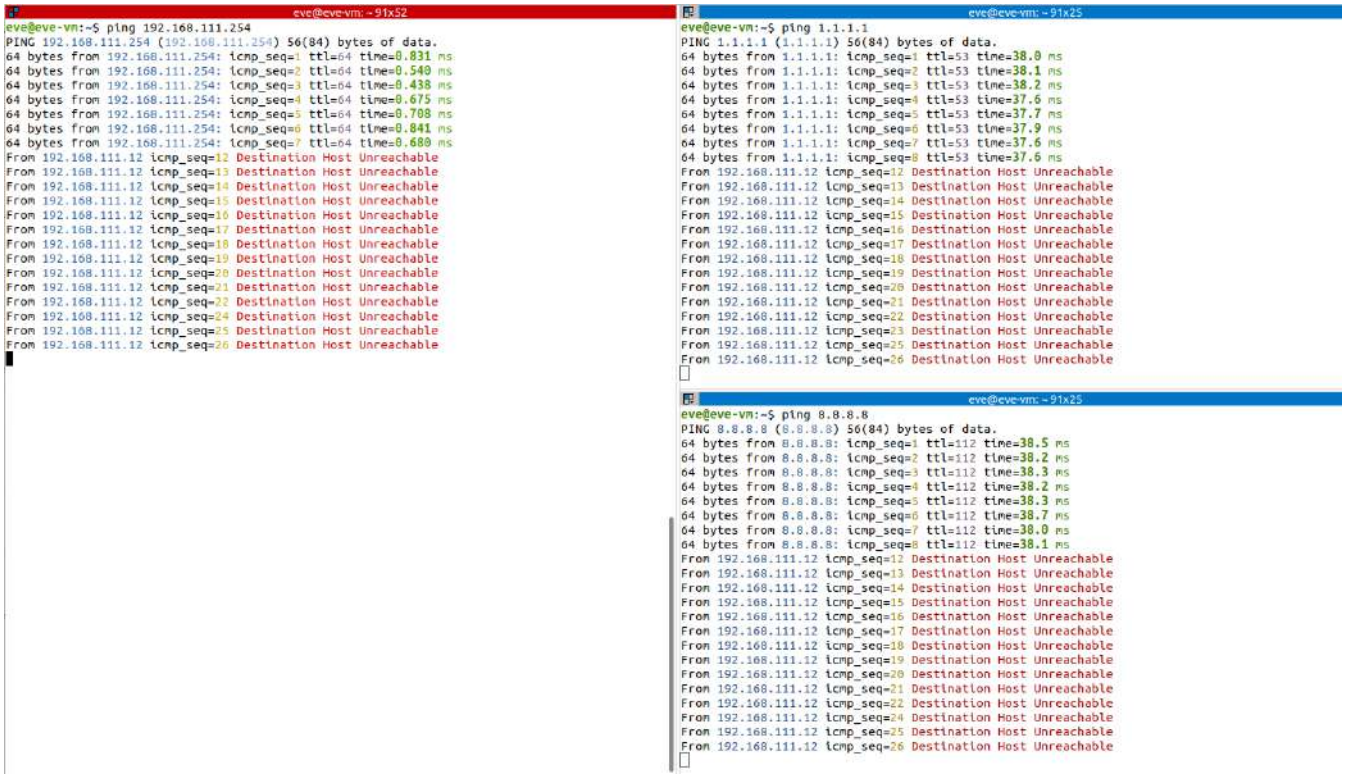


Figura 4.4: *Teste de ping para os destinos citados.*

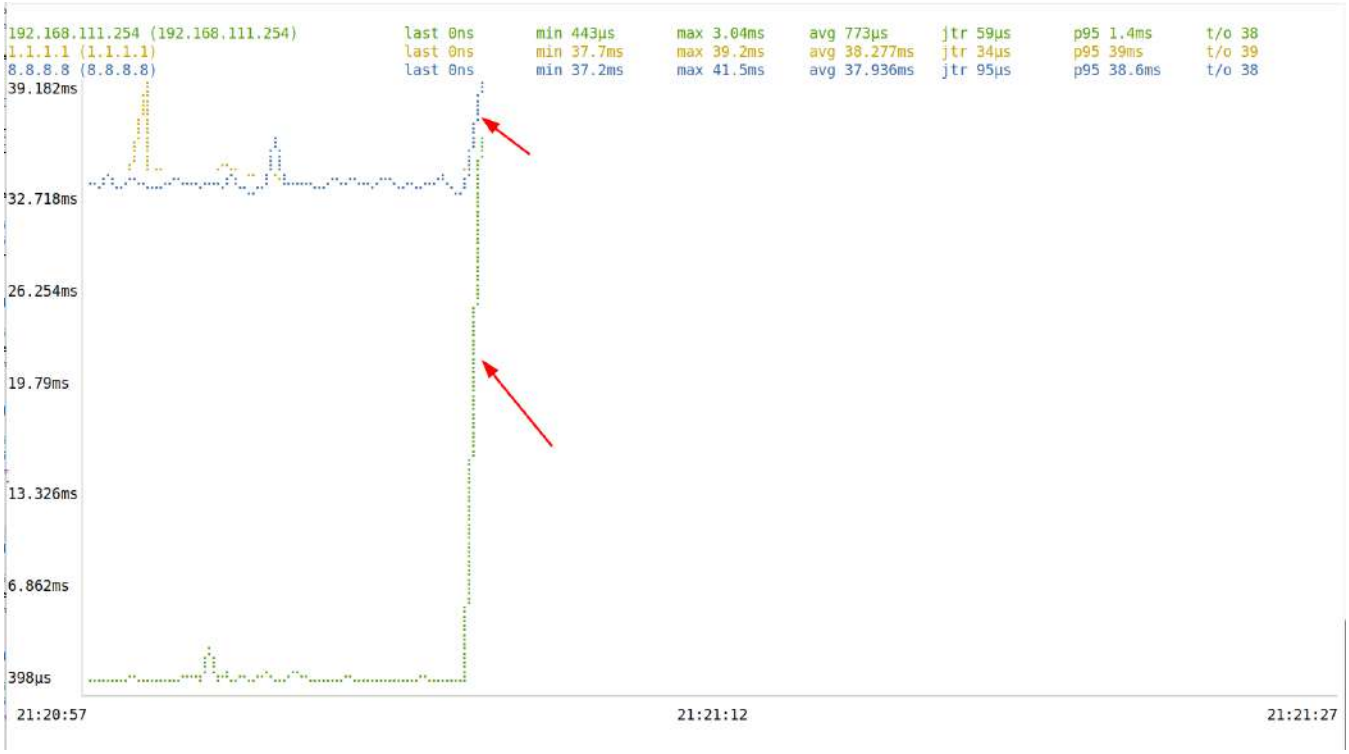


Figura 4.5: *Gráfico com o resultado do teste para os destinos citados.*

A perda pacotes ocorre de forma contínua, e com isso, todo o tráfego que passa pelo *firewall* é interrompido. Assim, o reestabelecimento dos serviços depende da intervenção por parte da equipe responsável.

4.2 Resultados com a topologia proposta no trabalho

Foram utilizados os mesmos endereços de destino citados nos resultados anteriores. Como é possível observar nas Figuras 4.6 e 4.7, o tráfego flui através do *firewall* principal sem interrupções:

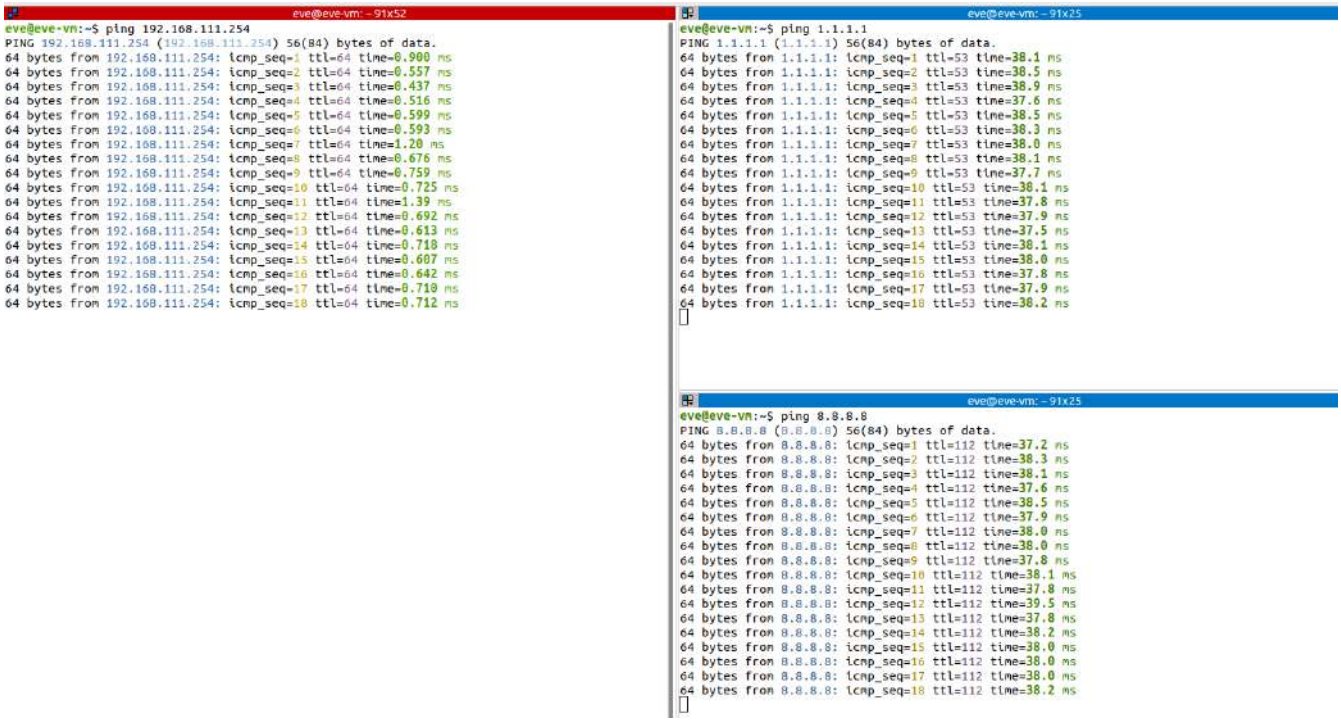


Figura 4.6: *Teste de ping para os destinos citados anteriormente.*

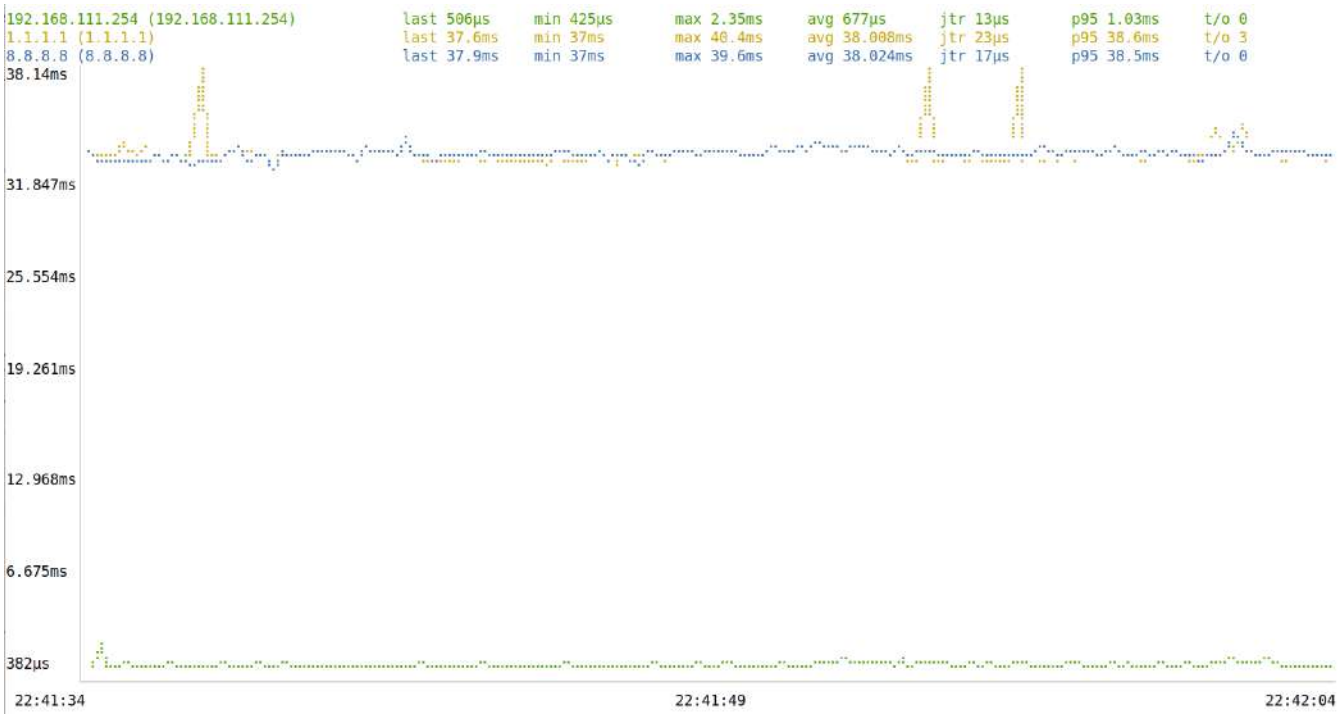


Figura 4.7: *Gráfico com o resultado do teste para os destinos citados anteriormente.*

Não há interrupções do tráfego para os destinos durante o teste e as latências médias são:

- **192.168.111.254** = 677 μs
- **1.1.1.1** = 38,008 ms
- **8.8.8.8** = 38.024 ms

A variação dos valores de latência em ambos os cenários, atual e proposto, foram mínimas. O foco principal desses testes foi analisar o comportamento do tráfego durante a interrupção do serviço de *firewall* e a atuação dos mecanismos de alta disponibilidade.

Já na Figura 4.8, é ilustrado o momento da interrupção do *firewall* principal (*master*), nesse momento, a falha é detectada pelo mecanismo de HA e o tráfego passa a ser encaminhado pelo *firewall* secundário (*backup*).

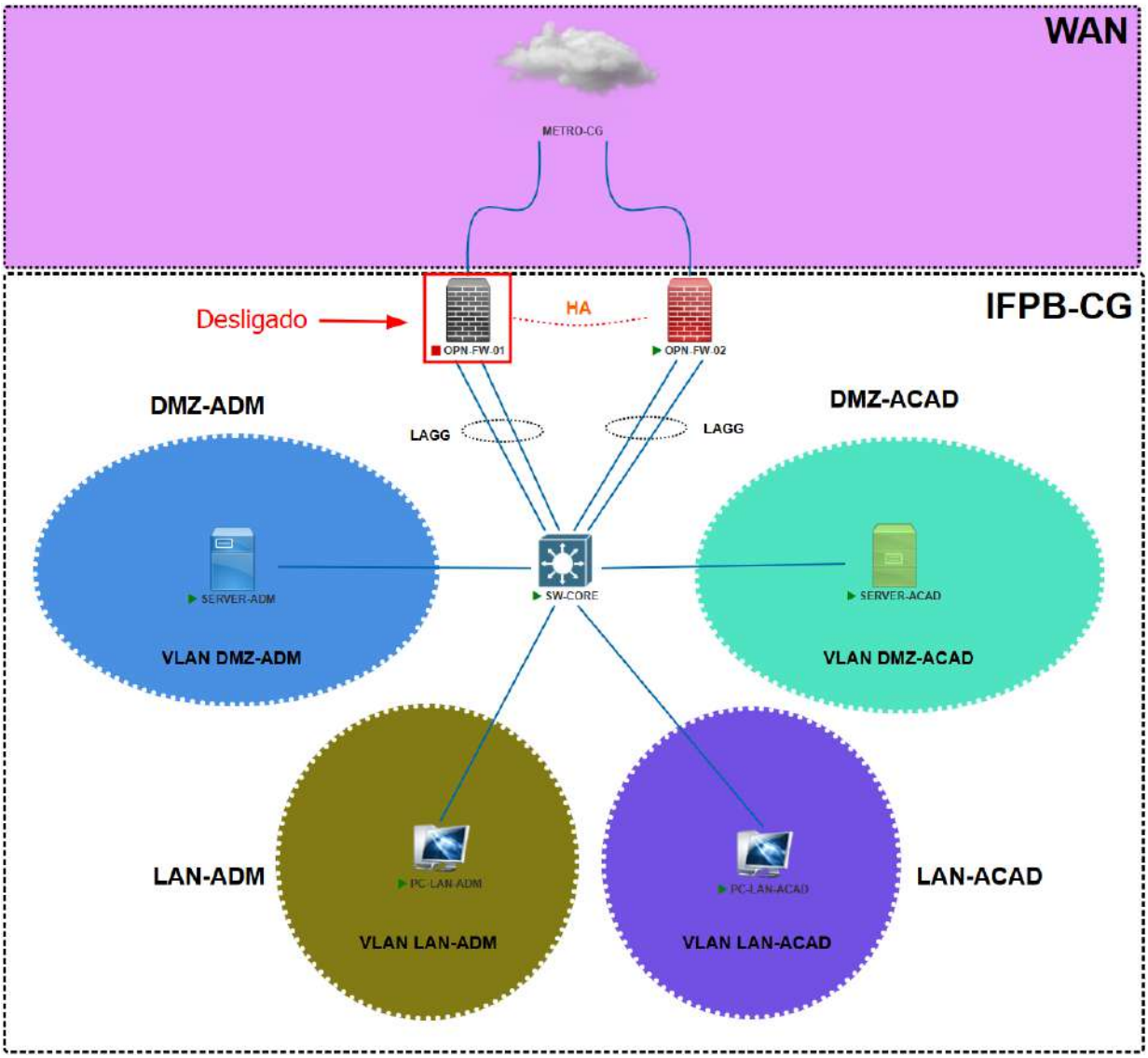


Figura 4.8: *Momento da interrupção do firewall master.*

Como pode se observar nas Figuras 4.9 e 4.10, há perda de pacotes na comunicação, entretanto, o tráfego é normalizado automaticamente. Nesse teste, foram perdidos, dois pacotes na comunicação com o endereço IP de uma das interfaces do *firewall* e três pacotes com os destinos externos. Essas perdas são consequência da falha no *firewall master* durante transição do tráfego para o *firewall backup*.

Figura 4.9: *Perda de pacotes detectada.*

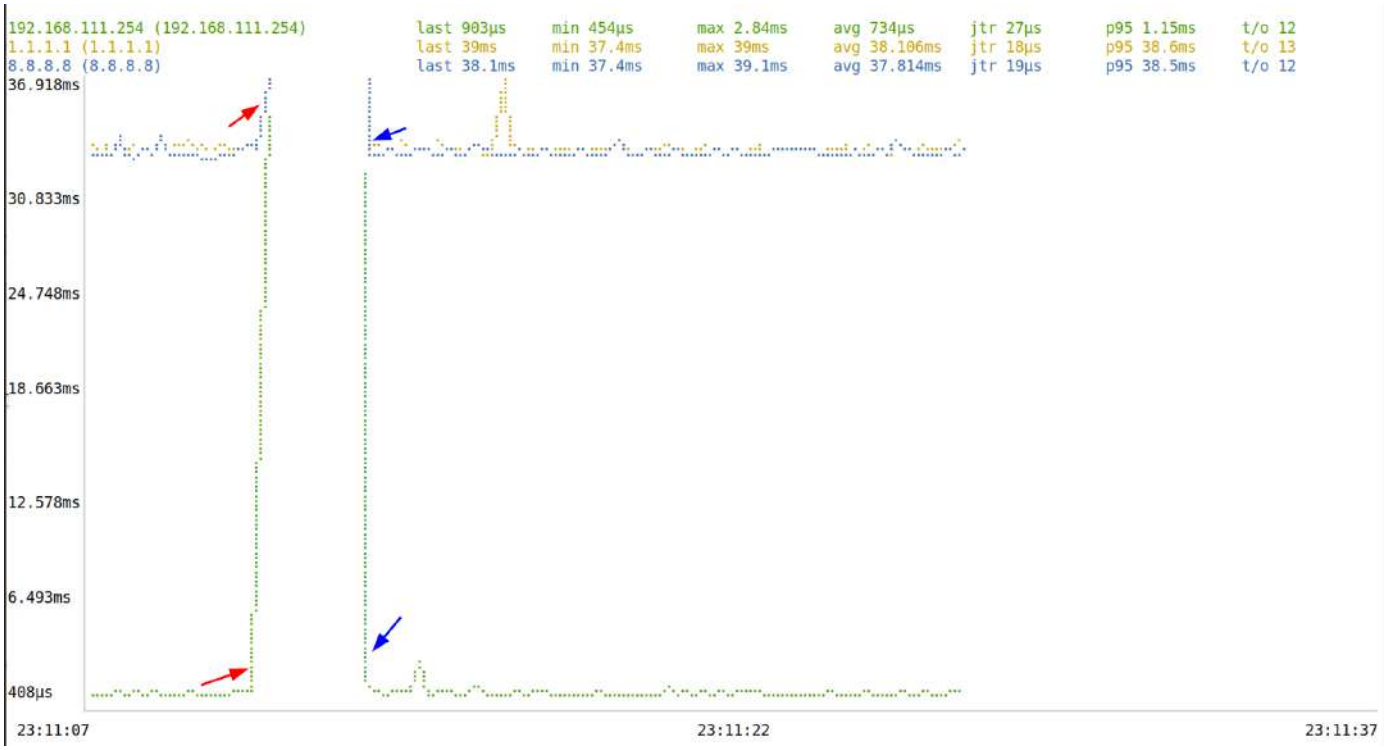


Figura 4.10: *Perda de pacotes detectada.*

Outra rodada de testes foi executada para avaliar o impacto da perda de pacotes, aplicando três diferentes intervalos de tempo na geração de tráfego. Dez ensaios foram executados para cada intervalo, com isso, foi possível ter uma ideia mais aproximada do comportamento do tráfego durante a transição entre os *firewalls*.

A Figura 4.11 ilustra o teste efetuado com o primeiro intervalo de tempo de um segundo (ou mil milissegundos – 1000ms) entre os pacotes:

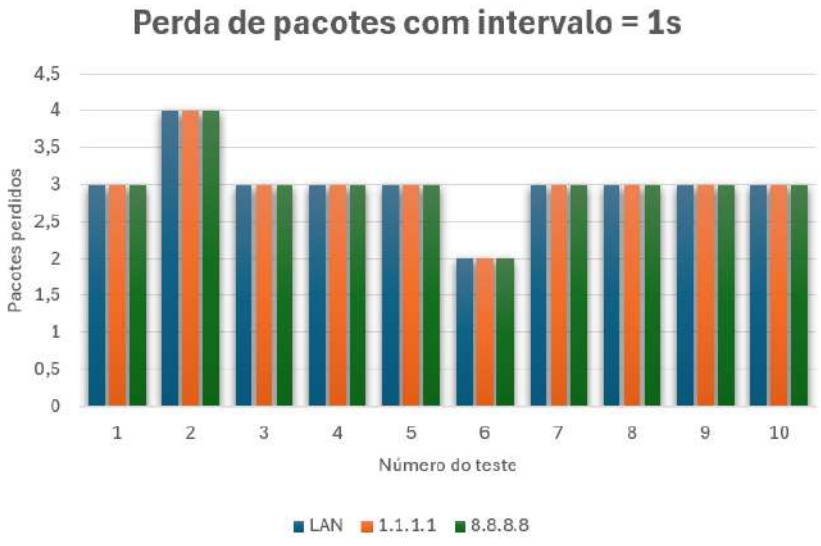


Figura 4.11: *Pacotes gerados com intervalo 1s.*

Já na Figura 4.12, pode-se observar o novo teste efetuado, agora, com um intervalo de 100ms entre os pacotes:

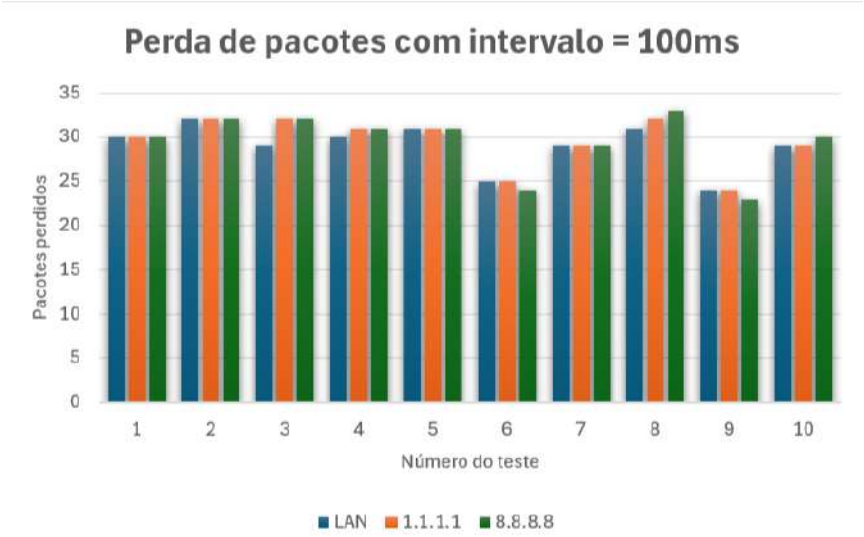


Figura 4.12: *Pacotes gerados com intervalo 100ms.*

Por fim, o resultado do ultimo teste pode ser visualizado na Figura 4.13:

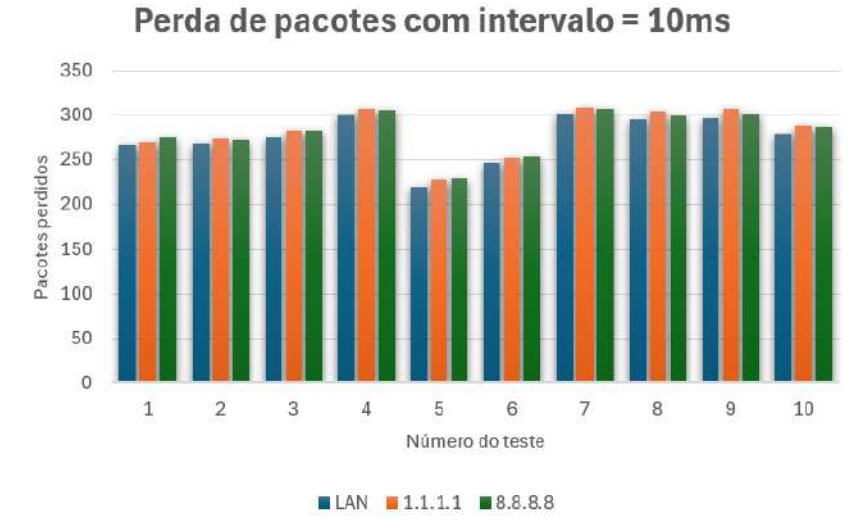


Figura 4.13: *Pacotes gerados com intervalo 10ms.*

As perdas de pacotes foram contabilizadas de forma absoluta, uma vez que, em valores percentuais, haveria uma variação, de acordo com a amostragem adotada. Nota-se que o número de pacotes perdidos aumenta conforme o fluxo de tráfego aumenta na rede, e é possível perceber também, que a interrupção do serviço do *firewall master*, leva cerca de 3 segundos até que o tráfego convirja para o *firewall backup*.

Como esperado, após a detecção de indisponibilidade do *firewall* principal, o secundário assume o papel de principal da rede, saindo do *status* de *backup* para *master* no CARP, conforme Figura 4.14:

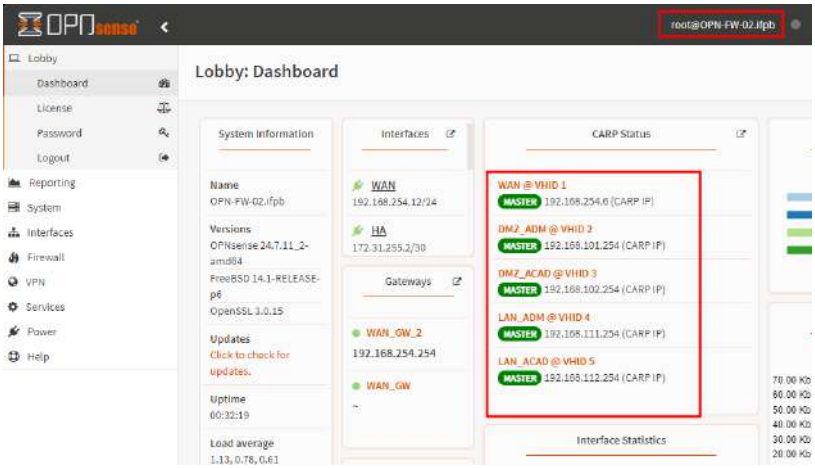


Figura 4.14: *Firewall backup assumindo o papel de master no CARP.*

Ele permanece com esse *status*, até que o *firewall* principal retorne à operação, depois disso, seu *status* volta para *backup*.

Capítulo 5

Considerações finais e sugestões para trabalhos futuros

Este trabalho teve como objetivo projetar uma solução de *firewall* com alta disponibilidade para o Instituto Federal de Educação, Ciência e Tecnologia da Paraíba (IFPB), *campus* Campina Grande. Durante seu desenvolvimento, foi possível entender o cenário atual da infraestrutura de redes e de segurança provida pelo, já defasado, sistema de *firewall* e estudar a aplicação de uma solução mais moderna e robusta.

Com a implementação do OPNsense, observou-se uma série de vantagens em relação à solução anterior, incluindo uma interface de gerência *WEB* mais intuitiva, maior facilidade de gerenciamento, integração nativa com diversas funcionalidades de segurança e um suporte ativo da comunidade. Além disso, a configuração de alta disponibilidade por meio do protocolo CARP, garantiu maior resiliência e minimizou o impacto de falhas no ambiente de rede.

O estudo da migração também demonstrou que a adoção do OPNsense pode proporcionar um aumento na eficiência da administração do *firewall*, reduzindo o tempo gasto com configurações complexas e aprimorando a visibilidade sobre os eventos de segurança da rede. No entanto, o sucesso desse processo depende de um planejamento adequado, envolvendo testes rigorosos e ajustes personalizados conforme as necessidades da infraestrutura existente.

Por fim, o estudo também reforça a importância da modernização das soluções de segurança para garantir a continuidade e a proteção dos serviços de rede. A substituição do iptables pelo OPNsense se mostrou uma alternativa viável e vantajosa, especialmente para ambientes que demandam alta disponibilidade e gerenciamento simplificado. Espera-se também que esse trabalho possa contribuir para a compreensão e aplicação das soluções propostas por outros estudiosos e profissionais.

5.1 Sugestões para trabalhos futuros

Como trabalho futuro, será realizada a etapa de implementação da solução proposta para validação e, depois disso, colocar a solução em operação no *campus*. Essa implantação precisou ser adiada, pelo fato do atraso na entrega dos equipamentos por parte dos fornecedores até a presente data da apresentação desse trabalho à banca avaliadora.

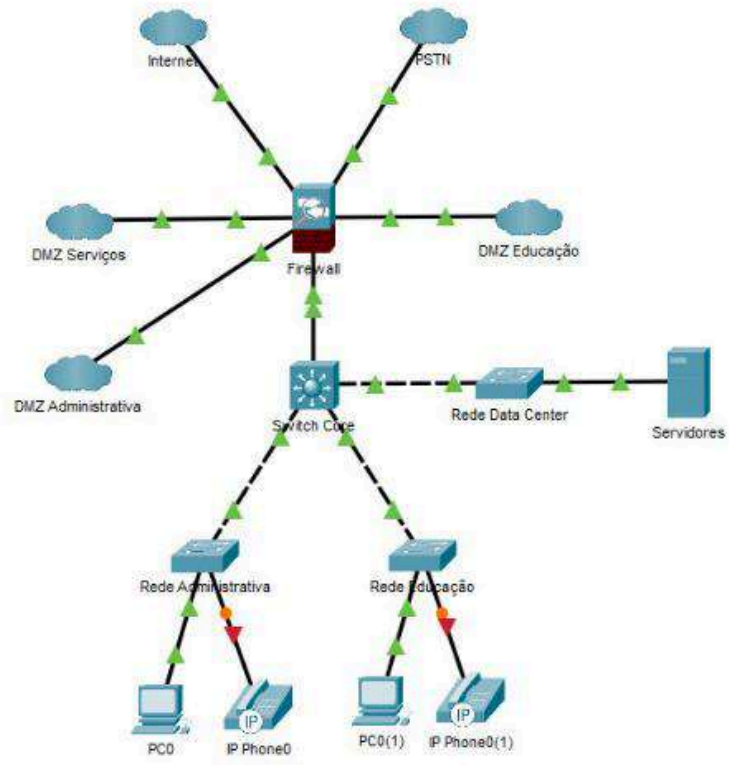
Por outro lado, com esse atraso, será possível também a realização de outro estudo para a possível adição de outras ferramentas agregadas a solução, tornando-a ainda mais eficiente.

Apêndice A

Questionário

Questionário para implantação de serviços de Firewall

1. **Informações Gerais do Projeto**
    a. Qual o objetivo principal da implementação do firewall?
    Resposta: Atualizar os recursos de segurança e serviços legados que comprometem a segurança da rede do campus.
    b. Qual a previsão de data para conclusão do projeto?
    Resposta: Até março de 2025.
    c. Existe um responsável designado pelo projeto? Se sim, informe o nome e o cargo.
    Resposta: Sim, Átila de Souza Medeiros, Técnico de tecnologia da informação responsável pela rede do campus.
2. **Infraestrutura de Rede**
    a. Quantos usuários/dispositivos utilizarão a rede protegida pelo firewall?
    Resposta: O NGFW a ser implantado deve suportar no mínimo 3000 usuários.
    b. Qual a topologia atual da rede? (Descreva brevemente ou forneça um diagrama).
    Resposta: A topologia da rede atual do campus, é ilustrada na figura:



A rede é dividida basicamente em 4 blocos: Administração, Educação, Data Center e DMZ, com suas respectivas separações em serviços administrativos, educacionais e serviços.

c. Qual a largura de banda atual da conexão de internet instalada?
   Resposta: 1Gbps.
d. Existem conexões redundantes ou backups de internet?
   Resposta: Não.
e. Quais as segmentações necessárias para as conexões das redes atendidas pelo firewall?
   Resposta: Basicamente, as mesmas ilustradas na topologia da rede.
f. Quais serviços precisam ser atualizados/migrados?

- ☑ ~~Firewall~~
- ☑ ~~DHCP~~
- ☑ ~~DNS~~
- ☑ ~~VPN~~
- ☑ ~~Servidor de Impressão~~
- ☑ ~~Servidor de Autenticação~~
- ☑ ~~Servidor WEB~~

3. **Políticas e Regras de Segurança**
   a. Quais serviços precisam ser permitidos (por exemplo, navegação, e-mail, FTP, VPN)?
      Resposta: Os serviços já conhecidos (80, 443, etc), além de algumas aplicações utilizadas pela comunidade.
   b. Existem regras de acesso específicas para diferentes departamentos ou grupos de usuários?
      Resposta: Sim, com ACLs de restrição já no switch core impedindo a comunicação entre redes administrativas e educacionais.
   c. Quais portas e protocolos precisam ser liberados ou bloqueados?
      Resposta: Precisa ser realizada uma análise minuciosa das regras já implementadas no IPtables para migração.
   d. Existem aplicações ou servidores críticos que precisam de regras personalizadas?
      Resposta: Sim.
   e. Há restrições de horário para o uso de determinados serviços ou aplicativos?
      Resposta: Não.
4. **Controle de acesso**
   a. Será necessário implementar autenticação de usuários para acesso à internet?
      Resposta: Sim. Precisa integrar o NGFW ao LDAP institucional.

b. Existem políticas específicas para dispositivos móveis e BYOD (Bring Your Own Device)?
   Resposta: Não.
c. Será necessário monitoramento ou controle de conteúdo web (filtragem de URLs)?
   Resposta: Sim, conforme legislação vigente.

5. **Recursos de VPN (Virtual Private Network)**
   a. Há necessidade de implementar VPN para acesso remoto?
      Resposta: Sim.
   b. Quantos usuários simultâneos usarão a VPN?
      Resposta: Aproximadamente 150 usuários.
   c. Qual tipo de VPN será utilizado (SSL, OpenVPN, Wireguard, IPSec, L2TP)?
      Resposta: Entre OpenVPN e Wireguard.
6. **Considerações Finais**
   a. Existe algum requisito adicional que não foi mencionado anteriormente?
      Resposta: Para uma melhor integração e controle da rede, o serviço DHCP deve ser migrado do *Switch Core* para o NGFW.
   b. Existe um orçamento definido para o projeto de implementação?
      Resposta: Não. Foi adquirido 2 servidores com as seguintes configurações:
      - 2x Intel Xeon Gold 6430;
      - 12x 32GB DDR5-5600 2Rx8 ECC RDIMM;
      - 2x NVME 480GB EM RAID1 Para o Sistema;
      - 4x 3,84TB, HOT-PLUG, CONSIDERANDO: TODOS OS DISCOS É DO MESMO MODELO E TER MESMA CAPACIDADE; PADRÃO SAS TRANSFERÊNCIA DE 12GB/S, OU SUPERIOR;
      - 8x 8TB (OITO TERABYTES), PADRÃO SAS, 12GBPS, HOTPLUG
      - 2x Placas SFP28 AOC-S25G-b2S com os devidos cabos
      - 2x Fontes PWS-1K68A-1R.

Referências Bibliográficas

AZIZ, M. Z. A. *et al.* Performance analysis of application layer firewall. In: *2012 IEEE Symposium on Wireless Technology and Applications (ISWTA)*. Bandung, Indonesia: Institute of Electrical and Electronics Engineers Inc., 2012. p. 182–186. 8

CARDELLINI, V. *et al.* Packet loss in data networks: Diagnosis and control. *IEEE Communications Surveys & Tutorials*, v. 12, n. 4, p. 1–15, 2010. Discussão sobre a perda de pacotes em redes de dados e técnicas para diagnóstico e controle. 15

DOCS-OPNSENSE. *OPNSense-Setup*. 2024. Accessado em 26 de Outubro de 2024. Disponível em: <https://docs.opnsense.org/setup.html>. 12, 23

GODOY, H. A. High availability em firewall utilizando o pfsense. *Sínteses: Revista Eletrônica do SIMTEC*, 2016. ISSN 2525-5398. 1, 4, 11

IEEE. *LAGG*. 2025. Accessado em 02 de Janeiro de 2025. Disponível em: <https://1.ieee802.org/tsn/802-1ax-rev/>. 13

JÚNIOR, M. A. B.; JÚNIOR, R. A. d. S. Relatório de migração de firewall virtual para appliance. 004, 2019. 4

KIRATSATA, H. J. *et al.* Behaviour analysis of open-source firewalls under security crisis. In: *2022 International Conference on Wireless Communications Signal Processing and Networking (WiSPNET)*. [S.l.: s.n.], 2022. p. 105–109. 2

KUROSE, J. F.; ROSS, K. W. *Computer Networking: A Top-Down Approach*. 7. ed. [S.l.]: Pearson, 2017. Este livro oferece uma introdução abrangente a redes de computadores, incluindo tópicos sobre perda de pacotes e protocolos de controle. ISBN 978-0133594140. 15

LARA, P. G. J. G. *Building Web Application Firewalls in High Availability Environments*. [S.l.: s.n.], 2010. 75-82 p. 4

LUTKEVICH, B. *Dmz (networking)*. [S.l.]: Retrieved from TechTarget: https://www. techtarget. com/searchsecurity . . . , 2021. 1

MONROSE, F.; WERNER, J. Tutorial: Parry and riposte: Honing cybersecurity skills with challenge-based exercises. In: *2018 IEEE Cybersecurity Development (SecDev)*. [S.l.: s.n.], 2018. p. 132–133. 2

MOTHERSOLE, I.; REED, M. J. *2011 Conference on Network and Information Systems Security*. [S.l.]: IEEE, 2011. ISBN 9781457707377. 6, 7

NETFILTER. *iptables*. 2025. Accessado em 12 de Janeiro de 2025. Disponível em: <https://www.netfilter.org/projects/iptables/index.html/>. 9

OPNSENSE. *OPNSense*. 2024. Accessado em 26 de Outubro de 2024. Disponível em: <https://opnsense.org/>. 1, 9

PNETLAB. *PNETLab*. 2024. Accessado em 26 de Outubro de 2024. Disponível em: <https://www.pnetlab.com/pages/documentation>. 12

REDHAT. *Redhat VM*. 2025. Accessado em 10 de Janeiro de 2025. Disponível em: <https://www.redhat.com/pt-br/topics/virtualization/what-is-a-virtual-machine>. 14

SHETH, C.; THAKKER, R. *2011 International Conference on Devices and Communications*. [S.l.]: IEEE, 2011. ISBN 9781424491902. 1

SOEWITO, B.; ANDHIKA, C. E. Next generation firewall for improving security in company and iot network. In: IEEE. *2019 International Seminar on Intelligent Technology and Its Applications (ISITIA)*. Jakarta, Indonesia, 2019. p. 205–209. 9

TRABELSI, Z. *et al.* Improved session table architecture for denial of stateful firewall attacks. *IEEE Access*, Institute of Electrical and Electronics Engineers Inc., v. 6, p. 35528–35543, 6 2018. ISSN 21693536. 7, 8

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA
Campus Campina Grande - Código INEP: 25137409
R. Tranqüilino Coelho Lemos, 671, Dinamérica, CEP 58432-300, Campina Grande (PB)
CNPJ: 10.783.898/0003-37 - Telefone: (83) 2102.6200

Documento Digitalizado Restrito

Entrega do trabalho de conclusão de curso

Assunto:	Entrega do trabalho de conclusão de curso
Assinado por:	Wesley Porto
Tipo do Documento:	Dissertação
Situação:	Finalizado
Nível de Acesso:	Restrito
Hipótese Legal:	Documento Preparatório (Art. 7o, $ 3o, da Lei no 12.527/2011)
Tipo do Conferência:	Cópia Simples

Documento assinado eletronicamente por:

- **Wesley Porto Santos**, **DISCENTE (202211210020) DE TECNOLOGIA EM TELEMÁTICA - CAMPINA GRANDE**, em 13/03/2025 15:49:22.

Este documento foi armazenado no SUAP em 13/03/2025. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse https://suap.ifpb.edu.br/verificar-documento-externo/ e forneça os dados abaixo:

Código Verificador: 1418138
Código de Autenticação: 6d6da0f43d