DISSERTAÇÃO DE MESTRADO

PROPOSTA DE ESQUEMA DE MONITORAMENTO E GERENCIAMENTO REMOTO DE REDES COMO PRESTAÇÃO DE SERVIÇO

DIEGO MARTINS DE OLIVEIRA

Brasília, janeiro de 2024

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

PROPOSTA DE ESQUEMA DE MONITORAMENTO E GERENCIAMENTO REMOTO DE REDES COMO PRESTAÇÃO DE SERVIÇO

PROPOSAL FOR A REMOTE NETWORK MONITORING AND MANAGEMENT SCHEME AS A SERVICE PROVISION

DIEGO MARTINS DE OLIVEIRA

ORIENTADOR: FÁBIO LÚCIO LOPES DE MENDONÇA, DR.
COORIENTADOR: RAFAEL TIMÓTEO DE SOUSA JÚNIOR, DR.

DISSERTAÇÃO DE MESTRADO PROFISSIONAL EM ENGENHARIA ELÉTRICA

PUBLICAÇÃO: PPEE.MP.064.
BRASÍLIA/DF: JANEIRO – 2024

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia

DISSERTAÇÃO DE MESTRADO

PROPOSTA DE ESQUEMA DE MONITORAMENTO E GERENCIAMENTO REMOTO DE REDES COMO PRESTAÇÃO DE SERVIÇO

DIEGO MARTINS DE OLIVEIRA

Dissertação de Mestrado submetida ao Departamento de Engenharia Elétrica como requisito parcial para obtenção do grau de Mestre em Engenharia Elétrica

Banca Examinadora

Prof. Fábio Lúcio Lopes de Mendonça, Dr. FT/UnB _____
Orientador/Presidente

Prof. Georges Daniel Amvame Nze, Dra. PPEE/UnB _____
Examinadora Interna

Prof. Raimundo Cláudio da Silva Vasconcelos,Dr.IFB _____
Examinador Externo

Prof. Robson de Oliveira Albuquerque, Dr. PPEE/UnB _____
Examinador/Suplente

FICHA CATALOGRÁFICA

DE OLIVEIRA, DIEGO MARTINS
PROPOSTA DE ESQUEMA DE MONITORAMENTOE GERENCIAMENTO REMOTO DE REDES COMO PRESTAÇÃO DE SERVIÇO [Distrito Federal] 2024.
xvi, 103 p., 210 x 297 mm (ENE/FT/UnB, Mestre, Engenharia Elétrica, 2024).
Dissertação de Mestrado - Universidade de Brasília, Faculdade de Tecnologia.
Departamento de Engenharia Elétrica

1. Gerenciamento de redes	2. Gerência remota
3. Serviços de rede	4. Arquitetura de rede
I. ENE/FT/UnB	II. Título (série)

REFERÊNCIA BIBLIOGRÁFICA

DE OLIVEIRA, D.M. (2024). *PROPOSTA DE ESQUEMA DE MONITORAMENTOE GERENCIAMENTO REMOTO DE REDES COMO PRESTAÇÃO DE SERVIÇO*. Dissertação de Mestrado, Departamento de Engenharia Elétrica, Publicação PPEE.MP.034, Universidade de Brasília, Brasília, DF, 103 p.

CESSÃO DE DIREITOS

AUTOR: DIEGO MARTINS DE OLIVEIRA
TÍTULO: PROPOSTA DE ESQUEMA DE MONITORAMENTOE GERENCIAMENTO REMOTO DE REDES COMO PRESTAÇÃO DE SERVIÇO.
GRAU: Mestre em Engenharia Elétrica ANO: 2024

É concedida à Universidade de Brasília permissão para reproduzir cópias desta Dissertação de Mestrado e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. Os autores reservam outros direitos de publicação e nenhuma parte dessa Dissertação de Mestrado pode ser reproduzida sem autorização por escrito dos autores.

DIEGO MARTINS DE OLIVEIRA
Depto. de Engenharia Elétrica (ENE) - FT
Universidade de Brasília (UnB)
Campus Darcy Ribeiro
CEP 70919-970 - Brasília - DF - Brasil

Agradecimentos

Em primeiro lugar, é claro agradeço a Deus por todas as oportunidades que recebi e que recebo até hoje.

Dedico este trabalho ao meu melhor amigo, meu porto seguro, meu ponto de partida. Se tivéssemos vivido mil anos juntos ainda seria pouco. Obrigado por tudo meu pai. Não tenho palavras para te descrever. Sentirei sua falta todos os dias pelo resto de minha vida.

Agradeço a minha esposa maravilhosa e minhas filhas lindas pela compreensão, apoio e pelo amor que dedicam a mim, sou muito grato por ter vocês na minha vida.

Agradeço ao meu orientador, professor Dr. Fábio Lúcio Lopes de Mendonça, que mais do que me orientar me incentivou e foi responsável direto pelo meu ingresso no mestrado, sem ele eu não teria chegado até aqui. De forma profissional e amiga nas horas mais complicadas durante este trabalho ele aturou nas dúvidas e problemas relativos e outros detalhes pertinentes à criação desta dissertação.

Ao meu Coorientador, professor Dr. Rafael Timóteo de Souza Júnior, que me ajudou de forma significativa na minha trajetória, um grande incentivador e apoiador dos emus estudos, com suas orientações e dicas.

Aos Professores do Departamento de Engenharia Elétrica, Georges Daniel Amvame Nze, Flávio Elias Gomes de Deus, Daniel Alves da Silva e Robson de Oliveira Albuquerque, pelas grandes dicas, constante apoio, incentivo e amizade, essenciais para o desenvolvimento deste trabalho. Agradeço também aos membros da banca.

Aos meus companheiros do Laboratório de Tecnologia da Tomada de Decisão – LATITUDE, da UnB, Kelly Oliveira, Wandembergue, Welber e todos os outros, pelo incentivo imprescindível.

Agradeço aos projetos a qual deram subsidio importantes para o desenvolvimento desta pesquisa. Agências brasileiras de pesquisa e inovação CNPq (Projeto INCT em Segurança Cibernética 465741/2014-2), CAPES (Projeto FORTE 23038.007604/2014-69) e FAPDF (Projetos UIoT 0193.001366/2016), bem como a Secretaria Especial do Desenvolvimento Social do Ministério da Cidadania, (Termo de Execução Descentralizada nº 01/2019 SNAS/MC), Projeto SISTER City – Sistemas Inteligentes Seguros e em Tempo Efetivo Real para Cidades Inteligentes (Secure near-real-time intelligent computing systems for Smart Cities (Edital 01/2020 - SEI:00193-00001250/2021-43) e ao Projeto AMORIS – Aplicativo Móvel e Central de Comando e Controle sobre Rede IoT para Suporte a Ações de Solidariedade no Combate ao COVID-19 e outros Surtos, Chamada Pública COPEI-DPI/DEX (Nº 01/2021 SEI n. 23106.104907/2021-05). Durante o desenvolvimento do trabalho, fui bolsista dos projetos TEDs (01/2019 DTIC),(01/2019 SNAS/MC),(01/2021) COPEI-DPI/DEX; agradeço às Instituições.

RESUMO

A dependência de recursos de Tecnologia da Informação e Comunicação por parte de pequenas e médias empresas tem crescido cada vez mais, as facilidades e oportunidades que esses recursos trazem se tonaram essenciais para as empresas. No Brasil, por exemplo, um simples vendedor capaz de aceitar o sistema de pagamentos eletrônico PIX, tem mais chances de vender do que outro que não utilize esse sistema. A dependência dos recursos de TIC vai dos menores negócios até médias e grandes empresas. No caso de pequenas e médias empresas, apesar da necessidade de se manter os recursos de TIC funcionando, manter uma equipe de manutenção local não é compatível com o porte da empresa, ao mesmo tempo, em que deixar para acionar um profissional apenas quando um problema já tiver acontecido, pode gerar um tempo de indisponibilidade alto e causar prejuízos financeiros. Neste sentido, a contratação de uma equipe externa para realizar o monitoramento dos recursos pode ajudar a prevenir problemas e reduzir o tempo de reação em caso de problemas.

Este trabalho tem em vista propor um esquema que envolve uma gama de *softwares* gratuitos, que possibilitem o monitoramento e gerenciamento remoto de redes e aplicações, de modo a permitir que uma equipe externa possa oferecer este monitoramento com serviço a outras empresas. Para atingir os objetivos foi preciso buscar por opções de soluções gratuitas não só para o monitoramento, mas também para contornar problemas de conexão ponto a ponto com IPv4 passando por CGNAT das operadoras. Foram criados cenários virtuais, físicos e mistos, com objetivo de efetuar testes de conexão, transmissão e estabilidade entre as redes, monitoramento de dispositivos e enlaces, envio de alerta, segurança e detecção de ataques. E por fim validar a proposta.

Após os testes e a análise dos resultados, conclui-se que é possível não só estabelecer conexão estável entre redes remotas sobre IPv4, sem custos adicionais com o provedor, como também, com um conjunto de soluções em *software* gratuitos, monitorar e gerenciar ativos e serviços em redes remotas. Possibilitando assim a utilização desses recursos para a criação de centrais de monitoramento remoto, capazes de oferecer seus serviços a várias redes clientes.

Palavras-chave: Monitoramento Remoto, Gerenciamento de Redes, Serviços de Rede.

ABSTRACT

The reliance on Information and Communication Technology resources by small and medium-sized businesses has grown more and more, and the facilities and opportunities that these resources bring have become essential for companies. In Brazil, for example, a simple vendor who is able to accept the PIX electronic payment system is more likely to sell than one who doesn't use this system. The dependence on ICT resources extends from the smallest businesses to medium-sized and large companies. In the case of small and medium-sized businesses, despite the need to keep ICT resources up and running, maintaining an on-site maintenance team is not compatible with the size of the company, while leaving it to a professional to be called in only when a problem has already occurred can lead to a high level of downtime and cause financial losses. In this sense, hiring an external team to monitor resources can help prevent problems and reduce reaction times in the event of problems.

The aim of this work is to propose an architecture involving a range of free software that enables remote monitoring and management of networks and applications, so that an external team can offer this monitoring service to other companies. In order to achieve the objectives, it was necessary to look for free solutions not only for monitoring, but also to get around point-to-point connection problems with IPv4 passing through the operators' CGNATs. Virtual, physical and mixed scenarios were created in order to carry out connection, transmission and stability tests between the networks, monitoring devices and links, sending alerts, security and detecting attacks. And finally to validate the proposal.

After testing and analyzing the results, it is concluded that it is possible not only to establish a stable connection between remote networks over IPv4, without additional costs from the provider, but also, with a set of free software solutions, to monitor and manage assets and services in remote networks. This makes it possible to use these resources to create remote monitoring centers, capable of offering their services to several client networks.

Keywords: Remote Monitoring, Network Management, Network Services.

SUMÁRIO

1 INTRODUÇÃO 1
1.1 MOTIVAÇÃO 2
1.2 OBJETIVOS DO TRABALHO 4
1.2.1 OBJETIVO GERAL 4
1.2.2 OBJETIVOS ESPECÍFICOS 4
1.3 TRABALHOS PUBLICADOS 4
1.4 METODOLOGIA 5
1.5 ORGANIZAÇÃO DO TRABALHO 5

2 REVISÃO BIBLIOGRÁFICA E TRABALHOS RELACIONADOS 6
2.1 IMPACTO DO ESGOTAMENTO IPV4 6
2.2 CARRIER GRADE NETWORK ADDRESS TRANSLATION CGNAT 6
2.2.1 PROBLEMAS COM CGNAT 8
2.3 RELEVÂNCIA DO CGNAT NESTE TRABALHO 9
2.4 ACESSO REMOTO 10
2.5 REDES VIRTUAIS PRIVADAS 11
2.6 ENDPOINT DETECTION AND RESPONSE - EDR 12
2.7 SOLUÇÕES EM *Software* 12
2.7.1 EMULADOR DE REDES GNS3 12
2.7.2 VIRTUALIZADOR QEMU 13
2.7.3 DOCKER 13
2.7.4 SERVIDOR *WEB* E SERVIDOR DE ARQUIVOS 14
2.7.5 PFSENSE *Firewall* 15
2.7.6 WIRESHARK 16
2.7.7 DD-WRT 16
2.7.8 ZABBIX 17
2.7.9 WAZUH 18
2.7.10 ZEROTIER 19
2.8 TRABALHOS RELACIONADOS 20

3 ESTRUTURA DA PROPOSTA DO ESQUEMA DE MONITORAMENTO 21
3.0.1 ZEROTIER 22
3.1 EQUIPAMENTO FÍSICO 23
3.2 EQUIPAMENTOS VIRTUAIS E EMULADOR 24
3.3 MÁQUINAS VIRTUAIS 25
3.4 CENÁRIOS 25
3.4.1 CENÁRIO - TOPOLOGIA BÁSICA 25

3.4.2 Cenário - Topologia maquinas físicas 26
3.4.3 Cenário - Esquema Final 26
3.5 Configurações 27
3.5.1 Configuração da rede virtual ZeroTier 27
3.5.2 Máquinas na rede Cliente 28
3.5.3 Máquinas na rede de serviços de monitoramento 37

4 Experimento e Resultados 45
4.1 Testes de rede 45
4.1.1 Teste de conexão 45
4.1.2 Testes de velocidade 46
4.2 Cenários e testes 48
4.2.1 Testes de rede na máquina hospedeira 48
4.2.2 Cenário 1 49
4.2.3 Cenário 2 56
4.2.4 Cenário 3 64
4.2.5 Cenário 4 73
4.3 Análise dos resultados 85
4.3.1 Testes de conexão e transmissão 85
4.3.2 Testes de Segurança 88

5 Conclusão 89
5.1 Trabalhos futuros 90

Referências Bibliográficas 91

6 Apêndices 94
6.1 Configurações 94
6.1.1 DD-WRT Virtualização 94
6.1.2 pFsense Zerotier 95
6.1.3 pFsense Wazuh 98
6.1.4 Wazuh Telegram 98
6.2 Testes 100
6.2.1 Cenário 2 100
6.2.2 Cenário 4 101

LISTA DE FIGURAS

2.1 (a) 7
2.2 (b) Comparação de arquitetura com NAT único (a) e com CGNAT (b). Adaptado de (SPALTER, 2022) 8
2.3 Saída do aplicativo nslookup. Os domínios consultados possuem apenas endereços IPv4. 9
2.4 Estatística do Google sobre adoção de IPv6 de seus usuários. Fonte: <https://www.google.com/intl/pt-BR/ipv6/statistics.html#tab=ipv6-adoption>, Acessado em 20/04/2023 10
2.5 Emulador GNS3 13
2.6 Comparação das camadas Docker e Máquina Virtual completa. Fonte: <https://www.docker.com/resources/what-container/>, Acessado em 21/04/2023 14
2.7 O Relatório mostra as tendências históricas no uso dos principais servidores da web desde maio de 2022. Fonte: <https://w3techs.com/technologies/history_overview/web_server>, Acessado em 21/04/2023 14
2.8 Recorte do ranqueamento de melhores soluções em *Firewall* apresentado pela plataforma PeerSport Fonte: <https://www.peerspot.com/categories/firewalls>, Acessado em 10/05/2023 15
2.9 Wireshark 16
2.10 DD-WRT 17
2.11 Comparação entre Zabbix e Nagios baseada em avaliações de usuários. Fonte: <https://www.gartner.com/reviews/market/infrastructure-monitoring-tools/compare/nagios-vs-zabbix>, Acessado em 25/05/2023 18
2.12 Arquitetura de componentes da plataforma Wazuh. Fonte: <https://documentation.wazuh.com/current/getting-started/architecture.html>, Acessado em 25/05/2023 ... 19

3.1 Etapas para construção do esquema da proposta. 21
3.2 Esquema básico de conexão com ZeroTier. 22
3.3 Esquema com *gateway* ZeroTier. 23
3.4 Topologia básica dos primeiros testes. 26
3.5 Topologia de testes com máquinas reais. 26
3.6 Topologia do cenário final de testes. 27
3.7 Lista de dispositivos conectados à rede virtual. 28
3.8 Gerenciador de Rotas na rede ZeroTier. 28
3.9 Detalhes da configuração de rotas e NAT do DD-WRT 30
3.10 Detalhes da configuração ZeroTier no pFsense parte 01 32
3.11 Detalhes da configuração ZeroTier no pFsense parte 02 33
3.12 Pacote Zabbix-agent no pFsense 34
3.13 Detalhes da configuração pFsense na rede de serviços de monitoramento. 38

3.14 Máquinas monitoradas pelo Zabbix. 39
3.15 Gráficos da máquina Xubuntu durante execução de um vídeo. 40
3.16 Gráficos da máquina Xubuntu depois da execução de um vídeo. 41
3.17 Mapa interativo da rede criado no Zabbix. 42
3.18 Alerta do Zabbix via Telegram. 42
3.19 Wazuh server - agentes e eventos de segurança. 43
3.20 Alerta do Wazuh via Telegram. 44

4.1 Cenário 1 Topologia - Testes de rede máquina hospedeira. 48
4.2 Cenário 1 Topologia - 2 redes virtualizadas na mesma máquina. 49
4.3 Cenário 1 teste de velocidade de acesso à *Internet*. 50
4.4 Cenário 1 Teste de traceroute da rede de serviços para a rede cliente. 51
4.5 Cenário 1 Teste de ping da rede de serviços para a rede cliente. 52
4.6 Cenário 1 Teste de transferência de arquivo com iPerf. 53
4.7 Cenário 1 janela de envio TCP saída iPerf. 54
4.8 Cenário 1 saída transferência com rsync. 55
4.9 Cenário 1 tela de monitoramento Zabbix. 56
4.10 Cenário 2 Topologia - 1 rede virtualizada e 1 rede física. 57
4.11 Cenário 2 - Foto rede virtualizada, máquina M1 com dois monitores 57
4.12 Cenário 2 - Foto rede com máquinas físicas 58
4.13 Cenário 2 - Foto rede com máquinas físicas 58
4.14 Cenário 2 Teste de traceroute da rede de serviços para a rede cliente. 59
4.15 Cenário 2 Teste de PING - Perda de pacotes. 60
4.16 Cenário 2 Teste de PING - RTT. 61
4.17 Cenário 2 Teste iPerf3 - Taxa de transmissão. 61
4.18 Cenário 2 Teste iPerf3 - Pacotes retransmitidos. 61
4.19 Cenário 2 saída transferência com rsync. 62
4.20 Cenário 2 tela de monitoramento Zabbix. 63
4.21 Cenário 2 acesso remoto Windows. 64
4.22 Cenário 3 Topologia - 2 redes físicas. 65
4.23 Cenário 3 - Foto rede de serviços física 65
4.24 Cenário 3 - Foto rede cliente com máquinas físicas 66
4.25 Cenário 3 - Teste de transferência com a *Internet* 67
4.26 Cenário 3 Teste de traceroute da rede de serviços para a rede cliente. 68
4.27 Cenário 3 Teste de PING - Perda de pacotes. 69
4.28 Cenário 3 Teste de PING - RTT. 69
4.29 Cenário 3 Teste iPerf3 - Taxa de transmissão. 69
4.30 Cenário 3 Teste iPerf3 - Pacotes retransmitidos. 70
4.31 Cenário 3 saída transferência com rsync. 71
4.32 Cenário 3 tela de monitoramento Zabbix. 72
4.33 Cenário 3 acesso remoto Windows. 73

4.34 Cenário 4 Topologia - 2 redes virtuais em 2 máquinas físicas diferentes. 74
4.35 Cenário 4 - Tela das redes virtualizadas 74
4.36 Cenário 4 - Teste de transferência com a *Internet* 75
4.37 Cenário 4 Teste de traceroute da rede de serviços para a rede cliente. 76
4.38 Cenário 4 Teste de PING - Perda de pacotes..... 77
4.39 Cenário 4 Teste de PING - RTT. 77
4.40 Cenário 4 Teste iPerf3 - Taxa de transmissão..... 77
4.41 Cenário 4 Teste iPerf3 - Pacotes retransmitidos..... 78
4.42 Cenário 4 saída transferência com rsync..... 79
4.43 Cenário 4 tela de monitoramento Zabbix. 80
4.44 Cenário 4 tela alerta de estado Zabbix/Telegram. 80
4.45 Cenário 4 Mapa de rede interativo Zabbix. 81
4.46 Estatística CERT.br, ataques por categoria. Fonte:https://stats.cert.br/incidentes/tipos-incidente 82
4.47 Cenário 4 alertas de varredura nmap..... 83
4.48 Cenário 4 tela de eventos de segurança Wazuh..... 83
4.49 Cenário 4 Teste de ataque de negação de serviço. 84
4.50 Cenário 4 Teste de ataque de negação de serviço, falha na página e detecção..... 84
4.51 Cenário 4 Teste de ataque de negação de serviço, falha na página e detecção..... 85
4.52 Resumo do resultado dos testes de PING..... 86
4.53 Resumo do resultado dos testes com iPERF. 86
4.54 Resumo do resultado dos testes com iPERF. 87

6.1 Versão do FeeBSD utilizada no pFsense..... 95
6.2 Arquivo de instalação zerotier-cli 96
6.3 Arquivo de instalação zerotier-gui 97
6.4 Arquivo de instalação wazuh-agent 98

LISTA DE TABELAS

2.1 Tabela de blocos de endereços IPv4 privados. 6

3.1 Máquinas fisicas. 24
3.2 Sistemas Operacionais e aplicações. 24
3.3 Máquinas virtuais recursos. 25

4.1 Elementos do Cenário 1. 49
4.2 Cenário 1 resultados PING. 52
4.3 Cenário 1 médias das métricas iPerf. 53
4.4 Cenário 1 resultados rsync. 55
4.5 Elementos do Cenário 2. 56
4.6 Cenário 2 médias das taxas iPerf3. 61
4.7 Cenário 2 resultados rsync. 62
4.8 Elementos do Cenário 3. 65
4.9 Cenário 3 médias das taxas iPerf3. 70
4.10 Cenário 3 resultados rsync. 71
4.11 Máquinas elementos do cenário 4. 74
4.12 Cenário 4 médias das taxas iPerf3. 78
4.13 Cenário 4 resultados rsync. 79
4.14 Resumo resultados dos testes de PING. 86
4.15 Resumo resultados dos testes de iPERF. 86
4.16 Resumo resultados dos testes de Rsync. 87

LISTA DE SIGLAS E ABREVIAÇÕES

ANATEL	Agência Nacional de Telecomunicações
API	*Application Programming Interface*
ARM	*Acorn Reduced Instruction Set Computer Machine*
CERT.BR	Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil
CGNAT	*Carrier Grade Network Address Translation*
CLI	*Comnand line interface*
CPU	*Central Processing Unit*
DDNW	*Dynamic Domain Name System*
DHCP	*Dynamic Host Configuration Protocol*
EDR	*Endpoint Detection and Response*
FreeBSD	*Free Berkeley Software Distribution*
FTP	*File Transfer Protocol*
ICMP	*Internet Control Message Protocol*
ISP	*Internet Service Provider*
LAN	*Local Area Network*
MIPS	*Microprocessor without interlocked pipeline stages*
MPEs	Micro e Pequenas Empresas
NAT	*Network Address Translation*
NIC.BR	Núcleo de Informação e Coordenação do Ponto BR
RTT	*Round Trip Time*
SIEM	*Security Information and Event Management*
SNMP	*Simple Network Management Protocol*
SSH	*Secure Shell*
TCP	*Transmission Control Protocol*
TIC	Tecnologia da Informação e Comunicação
UDP	*User Datagram Protocol*
URL	*Uniform Resource Locator*
VPN	*Denial of Service*
VPN	*Virtual Private Network*

1 INTRODUÇÃO

A utilização de recursos da Tecnologia da Informação e Comunicação (TIC), em Micro e Pequenas Empresas (MPEs) tem crescido e se tornado cada dia mais importante, segundo uma pesquisa da Fundação Getúlio Vargas publicada em 2021 (ABDI/FGV, 2021), que analisou o nível de maturidade digital das MPEs no Brasil, onde 18% das empresas que participaram da pesquisa estão no nível de maturidade considerado analógico, ou seja, não se utilizam de recursos de TIC, enquanto outras 48% estão ao nível emergente, 30% intermediário e 3% ao nível de líder digital.

Os recursos de TIC podem ser utilizados nas empresas das mais diversas formas, desde as mais simples e comuns como cadastros de clientes, controle de estoque, contabilidade, passando pela digitalização e armazenamento de documentos, contato com fornecedores, presença em mídias sociais, robôs de autoatendimento dentre outras inúmeras possibilidades. Não podemos esquecer também que recentemente passamos um período de isolamento devido à pandemia de COVID19, que acelerou a digitalização de alguns negócios que para sobreviver tiveram que migrar ou aumentar sua presença *on-line*.

Seja para realizar negócios *on-line*, contactar clientes ou simplesmente para viabilizar as mais diversas tarefas de seus colaboradores, o fato é que até pequenos negócios ou escritórios precisam ou podem se beneficiar de algum nível de informatização.

Ao mesmo tempo que a informatização traz vantagens e facilidades para as atividades do dia a dia de um negócio, ela também apresenta desafios, uma vez que determinada atividade passa a ser realizada por meio de recursos de TIC, como um computador e um ponto de acesso à rede interna ou à *Internet*, por exemplo, no caso de uma falha num equipamento ou no acesso às redes, as atividades que agora dependem desses recursos ficam prejudicadas. A gravidade do problema causado pela falha ou indisponibilidade de um desses serviços pode tomar proporções realmente prejudiciais ao negócio, chegando facilmente ao extremo de inviabilizar as atividades da empresa.

Não é difícil imaginar um caso em que uma falha num recurso de TIC possa inviabilizar as atividades de um negócio, basta pensarmos no caso de uma empresa que vende lanches via aplicativos de mensagens, se o equipamento de rede dessa empresa falha e ela fica sem acesso à *Internet*, os clientes não poderão enviar os pedidos e por tanto a empresa não fará vendas até que o problema na rede seja solucionado. Apesar de ser um caso simples, fica fácil entender o impacto negativo que uma falha nos recursos de TIC pode ter, até em micro empresas.

Mesmo com toda a sua importância, e estando presente em boa parte dos negócios hoje em dia, a área de TIC não é o foco principal da maioria das empresas que se beneficiam desses recursos, ou seja, elas apenas se utilizam desses recursos para atingir seus objetivos, e geralmente não existem profissionais de TIC no quadro de colaboradores da empresa. Nestes casos, como no exemplo

citado anteriormente, os problemas são geralmente detectados apenas quando um recurso para de funcionar e é necessário contratar o serviço de um profissional apenas para solucionar o problema.

Há ainda os casos onde, a depender das necessidades do negócio, a estrutura de TIC pode atingir uma complexidade que exija a presença de profissionais especializados para garantir a manutenção e o bom funcionamento dessa estrutura. Nestes casos se faz necessária a manutenção de uma equipe de TIC no local.

A criação e manutenção de uma equipe de TIC local, apesar de bem vinda, muitas vezes não é compatível como o porte da empresa, seja pelo fator financeiro ou até mesmo logístico. Uma possível saída nestes casos é a contratação de prestadores de serviços de TIC, que podem ser acionados em caso de necessidade de manutenção ou instalação de equipamentos, por exemplo. Por outro lado, essa necessidade de manutenção, na maioria dos casos, só é notada quando algo para de funcionar, o que pode se materializar em prejuízo, como já vimos anteriormente. O tempo de indisponibilidade do recurso será a soma do tempo gasto para acionar o profissional, mais seu tempo de deslocamento, mais o tempo para o diagnóstico e reparo do que for necessário, podendo ser ainda pior no caso de necessidade de substituição de peças.

O que nos leva a pensar não só na necessidade de reparo em caso de falha, mas também na necessidade de monitorar a estrutura e os serviços da rede local da empresa para uma possível antecipação de problemas ou mesmo uma reação mais rápida do profissional de TIC, que neste caso poderia identificar a falha antes de mesmo de ser acionado por alguém que esteja no local.

No mercado atualmente existem diversas soluções pagas de monitoramento e gerenciamento remoto de redes e serviços que prometem uma instalação e configuração facilitada, porém, ainda assim é necessário um profissional de TIC (administrador de rede), para realizar o trabalho e também para atuar quando uma falha for detectada, ou seja, serão dois custos, o custo da licença da solução de monitoramento e o custo do serviço do profissional que fará o monitoramento e as manutenções quando necessário.

Dessa forma, este trabalho objetiva propor e testar um esquema que envolve uma série de soluções em *softwares* gratuitos, eliminando os custos com licenças, e que permita que o monitoramento e gerenciamento remoto de redes e das aplicações nestas redes, seja ofertado como prestação de serviço por equipes especializadas, e que dentro do possível exija um mínimo de interferências ou modificações na infraestrutura da rede a ser monitorada por mais simples que ela seja.

1.1 MOTIVAÇÃO

A importância do monitoramento da rede e dos serviços hospedados nesta, vem do fato de que a indisponibilidade de algum recurso na rede pode levar não só a falhas no serviço, mas também podem se materializar em prejuízo financeiro. Haja visto que uma empresa que depende destes serviços para vendas ou outras transações financeiras, deve zelar pelo bom funcionamento de

sua estrutura de redes e serviços, pois em caso de indisponibilidade, existe o risco de prejuízos com as transações não realizadas durante a parada dos serviços, o que vai se somar ao custo da manutenção realizada para normalizar os serviços, ou seja, serão no mínimo dois prejuízos financeiros, além do prejuízo intelectual em caso de perda de dados.

Mesmo que não se trate de uma estrutura complexa, uma falha pode significar uma parada nas atividades do negócio e horas de trabalho perdidas. Micro e pequenas empresas que usam serviços de TIC como ferramenta para atingir seus objetivos, também devem se preocupar com a manutenção da sua estrutura informatizada, para evitar casos de indisponibilidade.

O monitoramento da rede e dos serviços pode ajudar a prevenir problemas com *softwares* e com equipamentos, ajuda também no dimensionamento destes recursos, uma vez que, quando se tem uma visão do que está em uso na estrutura, pode ser mais fácil mensurar o que está subutilizado e o que está sobrecarregado, além de alertar os responsáveis no caso de ocorrerem falhas que não seriam notadas rapidamente.

A manutenção de uma equipe de profissionais de TIC local para realizar o monitoramento e os reparos quando necessários, na maior parte dos casos não é compatível com o porte das MPEs, o que leva à contratação esporádica de profissionais apenas quando o problema já aconteceu, o que pode aumentar o tempo de resolução e o tempo de indisponibilidade.

Existem no mercado algumas soluções de monitoramento que prometem ser de fácil instalação e configuração, porém, suas licenças podem custar alguns milhares de reais, além disso, ainda é necessário a contratação de um profissional da área para agir quando um problema for encontrado, ou seja, ocorreria um gasto duplo.

Acreditamos que o monitoramento remoto, onde uma mesma equipe possa atender a várias empresas ao mesmo tempo, somando ao uso de soluções em *software* gratuito, possa criar uma opção de serviço de monitoramento e gerenciamento remoto de redes, com custo mais acessível a micro e pequenas empresas.

Neste sentido é necessário chegar a um conjunto que ofereça monitoramento e visualização do estado de ativos e dos enlaces, além de alarmes em caso de falhas, acesso remoto seguro para possíveis intervenções, análise da segurança dos dispositivos que pode ser realizada através de soluções *Security Information and Event Management* (SIEM).

Para proporcionar o monitoramento e o acesso remoto, é necessário adequar a solução às mais diversas condições de *links* ofertados por diversos provedores de *Internet* ou *Internet Service Provider* (ISP).

Em alguns casos, provedores locais, sobretudo os de menor porte, oferecem conexões IPv4 com *Carrier Grade Network Address Translation* (CGNAT), o que pode dificultar conexões ponto a ponto e por consequência, o estabelecimento de conexões remotas e *Virtual Private Network* (VPN)(NIC.BR, 2018). Isso nos leva a necessidade de estudar soluções para resolver os problemas sem a necessidade de trocar de ISP ou de contratar serviços extras, como endereçamento estático ou público, o que poderia gerar mais custos para a empresa cliente.

1.2 OBJETIVOS DO TRABALHO

1.2.1 Objetivo Geral

O objetivo principal deste trabalho é propor e testar um esquema que envolve um conjunto de soluções em *software* gratuito que viabilize a implementação de recursos de segurança, monitoramento e gerenciamento remoto de redes e aplicações, de modo que este possa ser ofertado como prestação de serviço por uma equipe especializada.

1.2.2 Objetivos Específicos

Para chegar no objetivo principal da proposta, foram elencados os seguintes objetivos específicos:

- Estudar e implementar solução para contornar problemas de conexão remota com IPv4 sem endereço IP estático direto do ISP;
- Testar aplicação de monitoramento de ativos de rede apontando para redes remotas;
- Testar aplicação de segurança com agentes em redes remotas;
- Testar o conjunto de soluções e verificar a capacidade de monitorar e gerenciar dispositivos em redes remotas, gerar alertas e realizar acessos remotos.

1.3 TRABALHOS PUBLICADOS

Durante a minha permanência no mestrado, conseguimos publicar 2 artigos que deram subsídios para os estudos aplicados nesta dissertação, de forma que o primeiro artigo *"Arquitetura para monitoramento e gerenciamento remoto de redes como prestação de serviços"* direcionado para o tema dessa dissertação abordando os temas de redes e de monitoramento remoto. O segundo artigo, *"aplicativo para avaliação de condução segura de usuários de veículos automotores por meio de inteligência artificial para benefícios em seguros veiculares"*, apesar de não está diretamente ligado ao tema da dissertação, trata-se de um artigo que utiliza técnicas de monitoramento de redes, com inteligência artificial para avaliação a condução de usuários de veículos automotores.

- OLIVEIRA, D. M. de; FILHO, F. D. C.; MENDONCA, F. L.; NZE, G. D. A.; SILVA, D. A.; SOUSA., R. T. D. Arquitetura para monitoramento e gerenciamento remoto de redes como prestação de serviços. *Conferências Ibero-Americanas WWW/Internet e Computação Aplicada 2022, Vilamoura. Lisboa: IADIS Press*, 2022.
- CALDAS, E. de O.; OLIVEIRA, D. M. de; MARQUES, G. dos S.; DEUS, F. E. G. de; MENDONCA, F. L.; SOUSA., R. T. D. Aplicativo para avaliação de condução segura de

usuários de veículos automotores por meio de inteligência artificial para benefícios em seguros veiculares. *Conferências Ibero-Americanas WWW/Internet e Computação Aplicada 2023, Vilamoura. Lisboa: IADIS Press*, 2023.

1.4 METODOLOGIA

Para alcançar os objetivos desta dissertação, na fase inicial fizemos a identificação e caracterização do problema, além do levantamento da solução com um conjunto de *softwares*, em seguida a verificação das ferramentas que possibilitassem a virtualização dos cenários e os testes. Testes estes que foram realizados tanto em cenários reais, com máquinas físicas, como em cenários virtualizados.

Seguimos com a criação de cenários físicos e virtuais, para testes de desempenho e estabilidade da conexão remota. Testes com solução de VPN capaz de contornar os problemas causados pelo *Network Address Translation* (NAT) duplo do ISP. Instalação e configuração de aplicações para monitoramento e gerenciamento com agentes nos dispositivos finais em redes remotas. Implantação de solução com recursos de segurança. Análise dos resultados obtidos e validação do esquema proposto.

1.5 ORGANIZAÇÃO DO TRABALHO

Este trabalho é composto por cinco capítulos, incluso este primeiro de introdução.

O segundo capítulo trata das tecnologias e técnicas utilizadas para gerenciamento de redes, monitoramento de dispositivos finais, implantação de recursos de segurança nos dispositivos, os conceitos do que foi aplicado e dos desafios encontrados.

O terceiro capítulo apresenta implantação do esquema proposto, as instalações e configurações realizadas, a disposição dos ativos de rede. Também é neste onde serão apresentados os diferentes cenários utilizados nos testes.

O quarto capítulo apresenta os testes realizados em cada cenário, físico e virtual, e a análise dos resultados obtidos.

O quinto capítulo conclui este trabalho trazendo a consolidação dos resultados e a validação da proposta, além da indicação de trabalhos futuros.

2 REVISÃO BIBLIOGRÁFICA E TRABALHOS RELACIONADOS

Neste capítulo estão presentes o conjunto de soluções e utilizadas para alcançar os objetivos da proposta. Alguns conceitos estudados também são apresentados para auxiliar o entendimento das tecnologias, que irão apoiar na execução do trabalho, podendo assim garantir a possibilidade de replicação dos experimentos e simulações utilizadas.

2.1 IMPACTO DO ESGOTAMENTO IPV4

O endereçamento IPv4 consiste em uma sequência de 32 bits utilizado para identificar cada dispositivo que faz parte de uma rede. Com estes 32 bits é possível gerar 4.294.967.296 de endereços. Ainda na década de 1990, ficou claro que esta quantidade de endereços não seria suficiente para comportar todos os dispositivos que acessam a *Internet*. Para amenizar o problema da falta de endereços IPv4 o *Network Address Translation* (NAT) foi criado (RFC2663, 1999)

Com a implementação do NAT, três faixas de endereços foram reservadas para uso em redes locais e chamadas de endereços privados, que não podem ser trafegados na *Internet*, por poderem se repetir em várias redes locais pelo mundo. Os blocos podem ser vistos na tabela 2.1, enquanto que os endereços que podem trafegar na *Internet* são conhecidos como endereços públicos. (RFC1918, 1996)

Início	**Fim**	**Máscara bits**	**Máscara decimal**	**Quantidade de endereços**
10.0.0.0	10.255.255.255	/8	255.0.0.0	$2^{24} = 16.777.216$
172.16.0.0	172.31.255.255	/12	255.240.0.0	$2^{20} = 1.048.576$
192.168.0.0	192.168.255.255	/24	255.255.255.0	$2^{8} = 65.536$

Tabela 2.1: Tabela de blocos de endereços IPv4 privados.

2.2 CARRIER GRADE NETWORK ADDRESS TRANSLATION CGNAT

Em sua implementação mais simples, o NAT permite que vários endereços privados sejam traduzidos para apenas um endereço público. Até a década de 2010 os ISPs usavam essa configuração para economizar os endereços públicos, fornecendo apenas um endereço público por cliente. Esse endereço é alocado no roteador de saída da rede cliente e na rede interna são utilizados endereços privados. Quando um dispositivo na rede cliente precisa acessar a *Internet*, antes de um quadro de requisição ser encaminhado para fora da rede local, o endereço de origem

privado é substituído pelo endereço público do roteador de saída, que mantêm registro de todas as trocas feitas pelo NAT, utilizando o endereço IP de origem e porta de origem para identificar cada troca, quado a resposta ao quadro chega, o roteador desfaz a troca, colocando de volta o endereço privado. (COMER, 2016).

Este esquema funcionou bem durante um tempo, porém com o crescente número de dispositivos que acessam a *Internet* e de clientes contratando serviços de acesso, os provedores começaram a ter dificuldade de fornecer endereços privados aos seus clientes, mesmo que apenas um por cliente.

Foi criado então o *Carrier Grade Network Address Translation* CGNAT, que consiste em aplicar o NAT ao nível de operadora, ou seja, um ISP pode utilizar um mesmo endereço IPv4 para um conjunto de clientes e aplicar um segundo NAT antes de sair da rede do ISP.

Para viabilizar essa técnica, a Autoridade para Atribuição de Números da *Internet*, do inglês *Internet Assigned Numbers Authority* IANA, instituição responsável pela distribuição global de endereços IP, reservou uma faixa de endereços IPv4, para serem utilizados no CGNAT, a faixa chamada de *Shared Address Space* é o bloco 100.64.0.0/10, que vai de 100.64.0.0 até 100.127.255.255 (RFC6598, 2012). A figura 2.2 (a) e (b), apresenta a diferença entre uma rede com NAT puro e uma com NAT duplo/CGNAT.

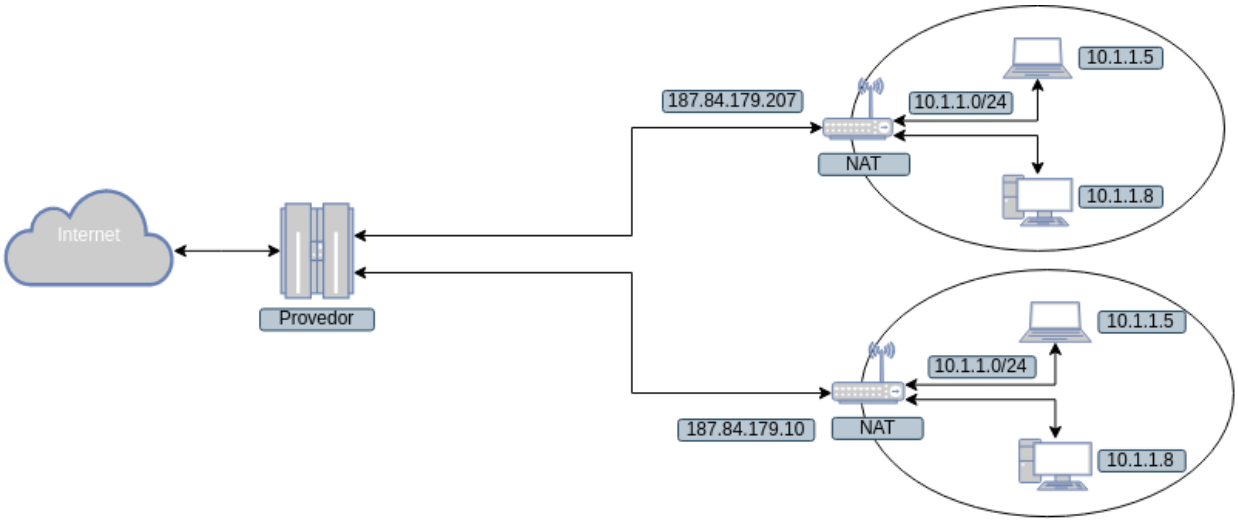


Figura 2.1: (a)

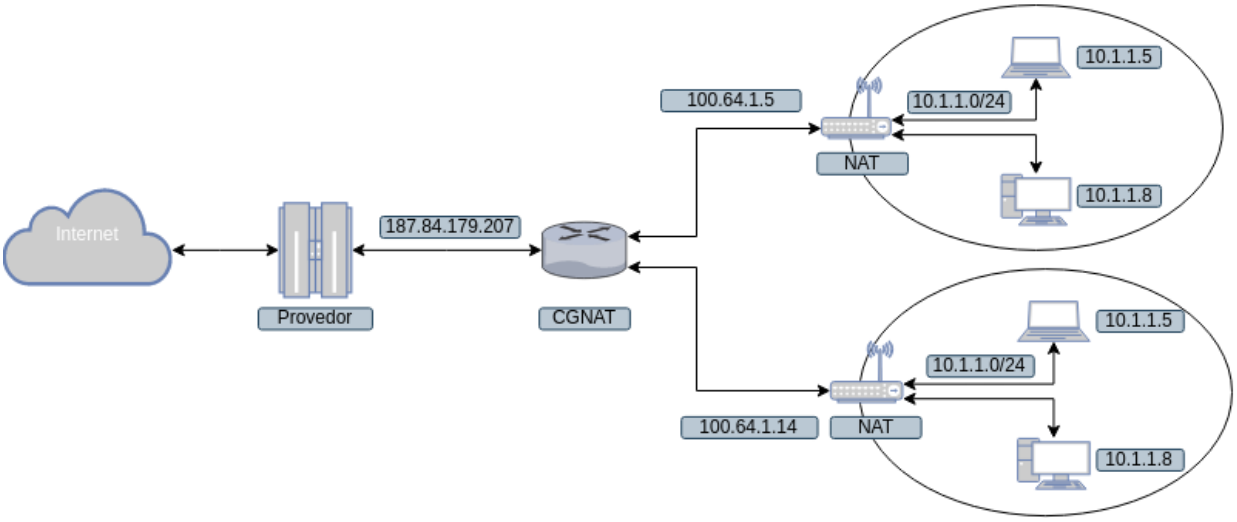


Figura 2.2: (b)
Comparação de arquitetura com NAT único (a) e com CGNAT (b). Adaptado de (SPALTER, 2022)

Apesar da existência de toda uma documentação das instituições reguladoras, o uso do bloco *Shared Address Space*, é apenas uma recomendação de boas práticas e existem relatos de ISPs usando outras faixas de endereços privados em suas configurações de CGNAT.

2.2.1 Problemas com CGNAT

Na tradução NAT o IP de origem e a porta de origem do quadro são alterados. O NAT único, realizado apenas na borda da rede cliente, pode causar problemas de conexão com determinados serviços, devido à troca de portas no caminho do quadro entre cliente e servidor. Para tentar resolver isso, é possível configurar algum redirecionamento de portas no roteador de borda do cliente, apesar de nem sempre ser suficiente para resolver o problema.

No CGNAT, uma tradução NAT é feita na borda da rede cliente e outra é feita na borda da rede do ISP, como apresentado na figura 2.2, ou seja, endereço e porta de origem são alterados duas vezes, além disso, há o compartilhamento de endereço IP entre clientes, o que pode aumentar a probabilidade de problemas de conexão.

Segundo (NIC.BR, 2018) o CGNAT fere um dos princípios da arquitetura da *Internet* que é a conexão ponto a ponto, cliente-servidor. Ainda pode interferir em aplicações como *peer-to-peer*, voz sobre IP, *streaming* de vídeo, jogos *on-line*, tunelamento de conexões, aplicações que utilizem endereço IP de origem como identificação do usuário, dentre outras.

Outra preocupação citada pelo artigo do NIC.br é da dificuldade de identificar o usuário pelo endereço IP, o que pode ter impacto em investigações criminais. Em (ANATEL, 2014) a Agência Nacional de Telecomunicações, ANATEL, prevê essa complicação e salienta a necessidade de que os ISPs e os provedores de conteúdo mantenham registros de conexões como informações de portas e registros de acessos às aplicações. Caso contrário, a impossibilidade de obter esses

registros pode ter impacto negativo em investigações criminais.

2.3 RELEVÂNCIA DO CGNAT NESTE TRABALHO

O CGNAT foi criado e ainda é utilizado como uma solução paliativa para o problema de esgotamento global do endereçamento IPv4, até a data de escrita deste trabalho, enquanto a implantação do endereçamento IPv6 ocorre.

De acordo com (NIC.BR, 2023) em matéria pulicada em seu *site* em 6 de abril de 2023, cerca de 45% da *Internet* no Brasil funciona com IPv6, porem ainda faltam equipamentos que suportem o protocolo. Além dos dispositivos, segundo levantamentos do NIC.br, apenas 8,7% dos bancos, 8,3% dos jogos *on-line*, 56,6% das redes sociais avaliadas possuem suporte à IPv6. O pior caso é dos *sites* do governo federal, onde quase todos ainda não tem suporte à IPv6. Utilizando a ferramenta nslookup para consultar domínios, foi possível verificar que *sites* como da Receita Federal, Universidade de Brasília e Instituto Federal de Brasília não tem endereços IPv6 vinculados a seus domínios ainda, como pode ser visto na Figura 2.3.

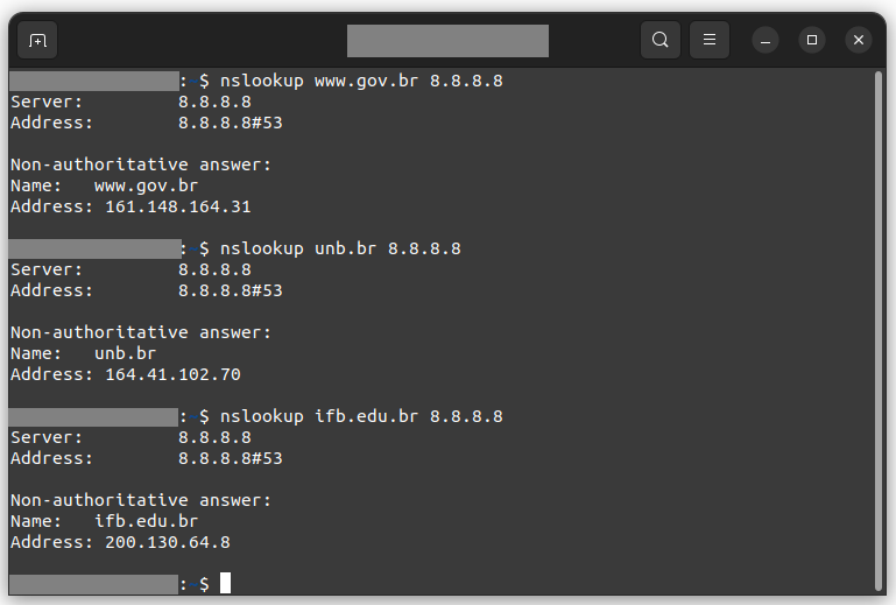


Figura 2.3: Saída do aplicativo nslookup. Os domínios consultados possuem apenas endereços IPv4.

Existem ainda provedores regionais de menor porte que não oferecem IPv6 para seus clientes. O Google também possui um *site* com estatísticas de adoção de IPv6 de seus usuários, que até a data da escrita deste trabalho estava próxima dos 45%, como pode ser visto na figura 2.4.

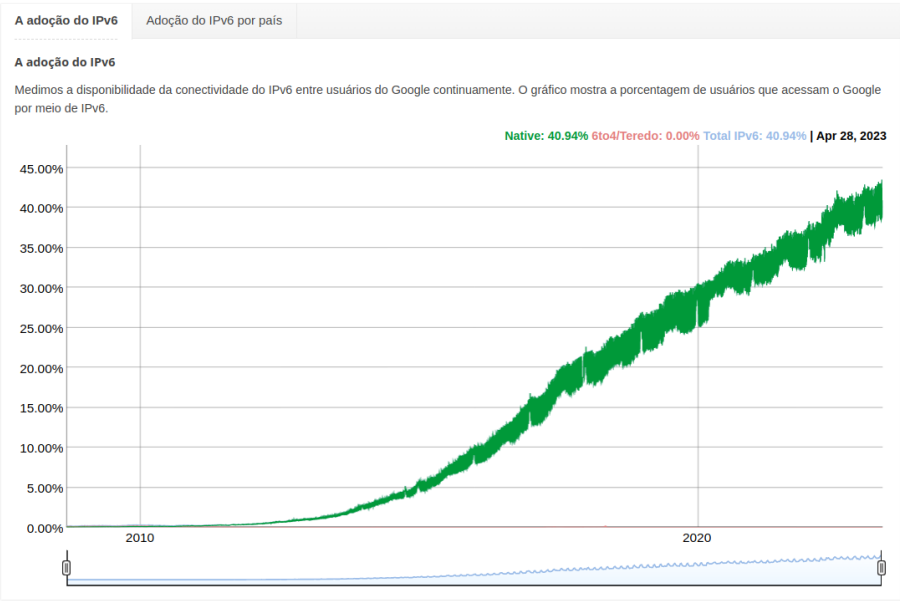


Figura 2.4: Estatística do Google sobre adoção de IPv6 de seus usuários. Fonte: <https://www.google.com/intl/pt-BR/ipv6/statistics.html#tab=ipv6-adoption>, Acessado em 20/04/2023

Com essas informações conclui-se que apesar de crescente, a adoção de endereçamento IPv6 ainda não atinge a maioria dos usuários da *Internet*, nem mesmo no Brasil, assim sendo, no sentido de tentar construir uma proposta que tenha um alcance maior, o endereçamento IPv6 não será utilizado nesta proposta, deixando essa abordagem para trabalhos futuros.

2.4 ACESSO REMOTO

O acesso remoto consiste basicamente acessar uma determinada máquina a partir de outra. Este acesso, em seu uso mais comum, se dá com objeto de administrar a máquina remota, emitindo comandos, alterando configurações dentre outas possibilidades. O acesso remoto a máquinas Linux pode ser realizado via linha de comandos, neste caso o protocolo mais utilizado é o *Secure Shell* SSH, com ele não só possível acessar outra máquina remotamente via linha de comandos, como também efetuar cópias de arquivos pela rede. Em ambientes, Windows o protocolo nativo de acesso remoto é o *Remote Desktop Protocol* RDP que provê acesso remoto com ambiente gráfico em sistemas operacionais Windows.

Quando as duas máquinas se encontram na mesma rede, a configuração pode ser relativamente simples, configuramos uma das máquinas para receber o acesso e a outra para realizar o acesso, além dessa configuração, é necessário saber o endereço IP ou nome da máquina a ser acessada.

Para acessar remotamente uma máquina em outra rede, não importando neste caso a distância geográfica, mas o simples fato de ser outra rede, é necessário haver roteamento entre essas redes.

Para acessar remotamente uma máquina em outra rede através da *Internet*, ainda é preciso saber o endereço da máquina de destino, e que haja roteamento, para que a informação saia da

origem passe pela *Internet* e cheque no destino. Devido ao esgotamento de endereços IPv4, o NAT é aplicado às redes locais e para passar pela *Internet* os endereços locais privados precisam ser traduzidos para pelo menos um endereço público, então o endereço IP usado para o acesso remoto será o endereço público do roteador da rede, agora é necessário configurar também o roteador de borda da rede, para encaminhar as requisições de acesso remoto para a máquina de destino.

Para entregar um IPv4 fixo as operadoras cobram taxas extras, e nem todas as operadoras oferecem IPv6. Existe então a questão de que os endereços públicos oferecidos pelas operadoras geralmente não são fixos, ou seja, o endereço muda, dificultando o acesso, o que poderia ser resolvido com o uso de um *Dynamic Domain Name System* DDNS e uma *Uniform Resource Locator* URL, que possibilita a atualização do endereço IP publico vinculado a URL.

O que se deseja neste trabalho é acessar remotamente todas as máquinas de uma rede e não apenas uma. Neste caso a melhor solução é a criação de uma VPN *site to site* entre as duas redes, que poderia ser feito utilizando DDNS e URL.

O problema é que devido ao uso de CGNAT pelas operadoras, um mesmo endereço público é compartilhado por vários clientes simultaneamente, e as portas são utilizadas para diferenciar as conexões, neste caso, para o DDNS funcionar corretamente é necessário solicitar à operadora que faça um redirecionamento de portas específicas.

Para contornar todos esses problemas e sem ter gastos e sem depender de configurações adicionais da operadora, optamos por utilizar o serviço gratuito da plataforma ZeroTier que permite criar uma rede virtual que pode ser utilizada para interconectar outras sub-redes passando pela *Internet*.

2.5 REDES VIRTUAIS PRIVADAS

Segundo (COMER, 2016), Redes Virtuais Privadas, do inglês *Private Virtual Network* VPN, é uma tecnologia amplamente utilizada para fornecer acesso seguro à rede interna de uma organização. Originalmente concebida para ser uma alternativa à utilização de *links* privados considerados financeiramente caros, a VPN permite estabelecer uma conexão ponto a ponto entre redes geograficamente distantes de modo seguro, por implementar criptografia na conexão, onde os quadros enviados pela *Internet*, considerada insegura, são encriptados na origem e desencriptados no destino, formando um túnel seguro entre as localidades.

Existem diferentes configurações de VPN com os mais diversos objetivos, desde acesso remoto à navegação anônima. Em suas configurações mais comuns, podemos destacar, a VPN de acesso remoto *Client to Site*, que permite, por exemplo, que um colaborador de uma empresa acesse a rede dessa empresa e seus recursos direto do próprio computador, mesmo que este esteja geograficamente distante, neste caso, na rede remota, é instalada uma máquina que funciona como servidor de VPN, enquanto no computador externo é instalado um *software* que irá se conectar e

autenticar na rede remota (KLUSAITė, 2020).

Já a VPN sítio a sítio *Site to Site* permite, por exemplo, interligar a rede de uma matriz à rede de uma filial através da *Internet*, neste caso nas duas redes serão instaladas máquinas que funcionam como pontes / *gateways* entre as redes, encaminhando quadros pelo túnel de uma rede à outra, assim, após a autenticação entre as duas pontas, as máquinas de uma rede terão acesso às máquinas na outra rede (KLUSAITė, 2020).

2.6 ENDPOINT DETECTION AND RESPONSE - EDR

EDR do inglês *Endpoint Detection and Response*, detecção e resposta de dispositivo final, segundo (KASPERSKY, 2023) é uma abordagem integrada em camadas para proteção de dispositivos finais que combina monitoramento em tempo real e análise dos dados com resposta baseada em regras. As soluções de EDR geralmente utilizam um agente instalado no dispositivo, que coleta dados e envia para uma entidade central que realiza o processamento e comparações basadas em regras, e quando necessário pode emitir alertas e outras ações.

2.7 SOLUÇÕES EM *SOFTWARE*

Para compor a proposta de gerenciamento e monitoramento remoto foi utilizado um conjunto de aplicações que permitem a coleta, armazenamento e análise de dados do conjunto de máquinas a ser monitorado, além de soluções em acesso remoto para viabilizar a coleta desses dados em uma rede remota.

Para realizar os testes em diferentes cenários e verificar a viabilidade da proposta, também foi utilizado um conjunto de aplicações de virtualização que abrangem deste a virtualização de sistemas operacionais e ativos de redes até redes completas.

2.7.1 Emulador de redes GNS3

O GNS3 é um *software* de emulação de redes, que permite criar e testar cenários de redes, além de conectar os cenários a *Internet* para testes mais realistas, a aplicação tem suporte a sistemas de vários fabricantes de equipamentos de rede, além de compatibilidade com vários virtualizadores de sistemas operacionais, todas essas características permitem a criação de cenários deste os mais simples até os mais complexos. A variedade de equipamentos suportados pelo emulador permite que seja usado para treinamentos e para testar configurações antes de implantar em redes reais. A aplicação pode ser executada localmente utilizado os recursos de *hardware* da máquina local, ou pode se conectar a um servidor com maiores recursos de processamento, permitindo a criação de cenários com mais dispositivos. (GNS3, 2023). Uma imagem da tela da ferramenta pode ser

vista na figura 2.5.

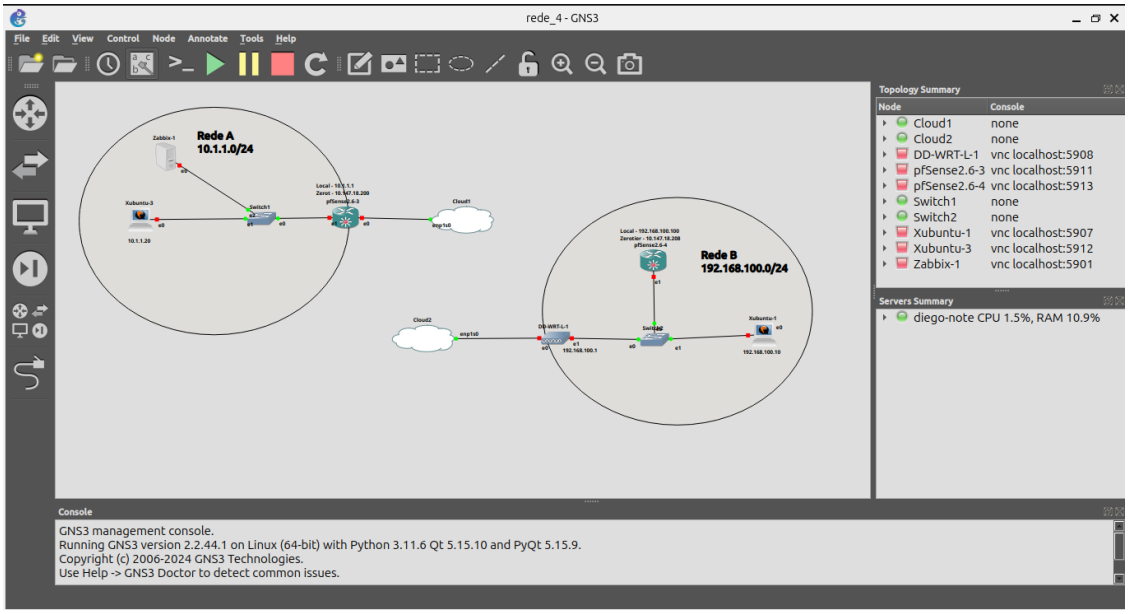


Figura 2.5: Emulador GNS3

2.7.2 Virtualizador QEMU

O QEMU é um *software* livre capaz de emular uma máquina com todos os recursos necessários para a instalação dos mais diferentes sistemas operacionais. Em seu modo *System Emutation*, o processador ou do inglês *Cenral Processing Unit* CPU pode ser completamente emulado, ou utilizando um *hypervisor* o sistema operacional virtualizado pode ser executado direto na CPU da máquina hospedeira, geralmente melhorando o desempenho. No modo *User Mode Emulation* o Qemu pode executar processos compilados para uma CPU diferente da usada na máquina hospedeira, como processadores *Acorn Reduced Instruction Set Computer Machine* ARM e *Microprocessor without interlocked pipeline stages* MIPS (QEMU, 2022).

O GNS3 suporta tanto Qemu quanto VirtualBox como solução de virtualização, porém com o VirtualBox é necessário criar uma máquina e um disco para cada instancia de equipamento a ser virtualizado, enquanto que com o Qemu é possível criar apenas um *template* de um equipamento e várias instancias do mesmo, que iram compartilhar o disco e mesmo assim manter diferentes configurações.

2.7.3 Docker

O Docker é uma solução que utiliza contêineres para executar aplicações virtualizadas ao nível de sistema operacional. Os contêineres contem tudo que a aplicação necessita para ser executada, bibliotecas, executáveis, ferramentas de sistemas, arquivos de configuração e outras dependências, eles formam ambiantes isolados onde a aplicação é executada. Como os contêineres são

isolados uns dos outros, vários podem ser executados na mesma máquina de foma independente, cada um com seu ambiente. Os contêineres compartilham o *kernel* com o sistema operacional, consumindo menos recursos que a virtualização de sistema operacional inteiro. Na figura 2.6 é possível ver uma comparação das camadas do Docker, uma máquina virtual completa (DOCKER, 2023).

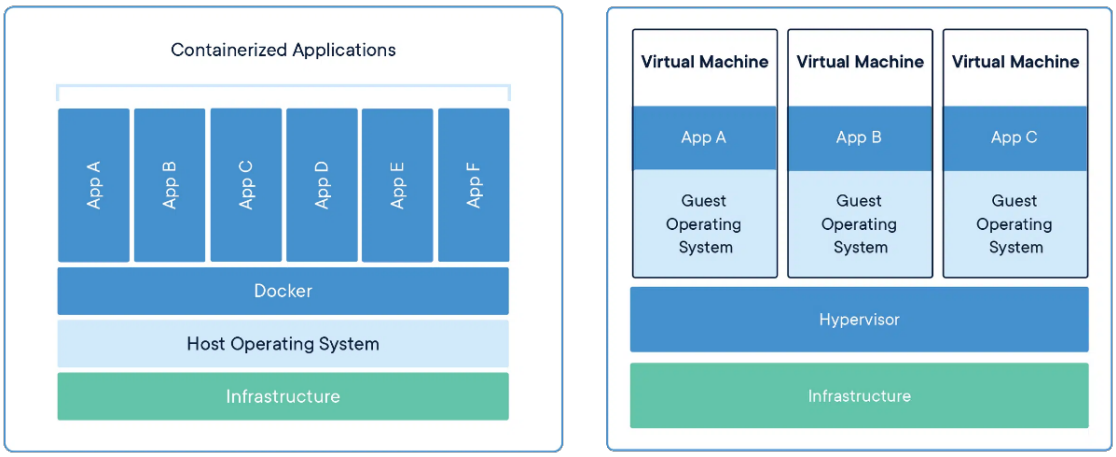


Figura 2.6: Comparação das camadas Docker e Máquina Virtual completa. Fonte: <https://www.docker.com/resources/what-container/>, Acessado em 21/04/2023

2.7.4 Servidor *WEB* e Servidor de arquivos

A Apache *Software Foundation* é uma fundação voltada para o desenvolvimento de *software* livre, responsável por mais de 350 projetos de código aberto, dentre eles o Apache HTTP *Server*, aplicação do tipo *WEB Server*, para criar e hospedar sites, de instalação e manutenção relativamente simples e uma vasta documentação (ASF, 2023). Segundo (W3TECHS, 2023) site de pesquisas sobre tecnologia, o Apache HTTP *Server* é o segundo *WEB Server* mais utilizado no mundo, em pesquisa realizada ente maio de 2022 e maio de 2023. O resultado da pesquisa pode ser visto na figura 2.7

	2022 1 May	2022 1 Jun	2022 1 Jul	2022 1 Aug	2022 1 Sep	2022 1 Oct	2022 1 Nov	2022 1 Dec	2023 1 Jan	2023 1 Feb	2023 1 Mar	2023 1 Apr	2023 1 May
Nginx	33.3%	33.5%	33.6%	33.8%	34.0%	34.3%	34.2%	34.0%	33.9%	34.0%	34.3%	34.5%	34.4%
Apache	31.6%	31.5%	31.4%	31.4%	31.3%	31.2%	31.4%	32.0%	32.8%	33.0%	32.3%	32.2%	32.1%
Cloudflare Server	21.5%	21.6%	21.6%	21.7%	21.7%	21.6%	21.6%	21.2%	20.5%	20.1%	20.2%	20.2%	20.4%
LiteSpeed	12.1%	12.1%	12.2%	12.3%	12.2%	12.3%	12.3%	12.1%	11.8%	11.6%	11.7%	11.8%	11.8%
Microsoft-IIS	6.0%	6.0%	6.0%	5.9%	5.9%	5.9%	5.9%	5.8%	5.8%	5.7%	5.7%	5.6%	5.6%
Node.js	1.9%	1.9%	2.0%	2.0%	2.1%	2.1%	2.1%	2.1%	2.0%	1.9%	2.0%	2.0%	2.0%

Figura 2.7: O Relatório mostra as tendências históricas no uso dos principais servidores da web desde maio de 2022. Fonte: <https://w3techs.com/technologies/history_overview/web_server>, Acessado em 21/04/2023

2.7.5 pfSense *Firewall*

O pfSense é um sistema operacional de distribuição livre, baseado no sistema *Free Berkeley Software Distribution* FreeBSD, que neste caso é customizado para funcionar como *firewall* e roteador. O sistema conta com interface *web* intuitiva e com diversos pacotes que implementam as mais diversas funcionalidades, além de contar com uma comunidade ativa e atualizações frequentes (FENCING, 2023).

Segundo a plataforma PeerSpot (PEERSPOT, 2023), que publica análises e comparações de tecnologias, em publicação de abril de 2023 entre as soluções de *firewalls* pesquisadas, o pfSense ficou em terceiro lugar dentre os mais utilizados, ficando atrás apenas do Fortnet Fortigate e Cisco Secure Firewall, duas soluções pagas, onde é necessário comprar o equipamento físico e a licença de *software*. Um recorte da lista pode ser visto na figura 2.8.

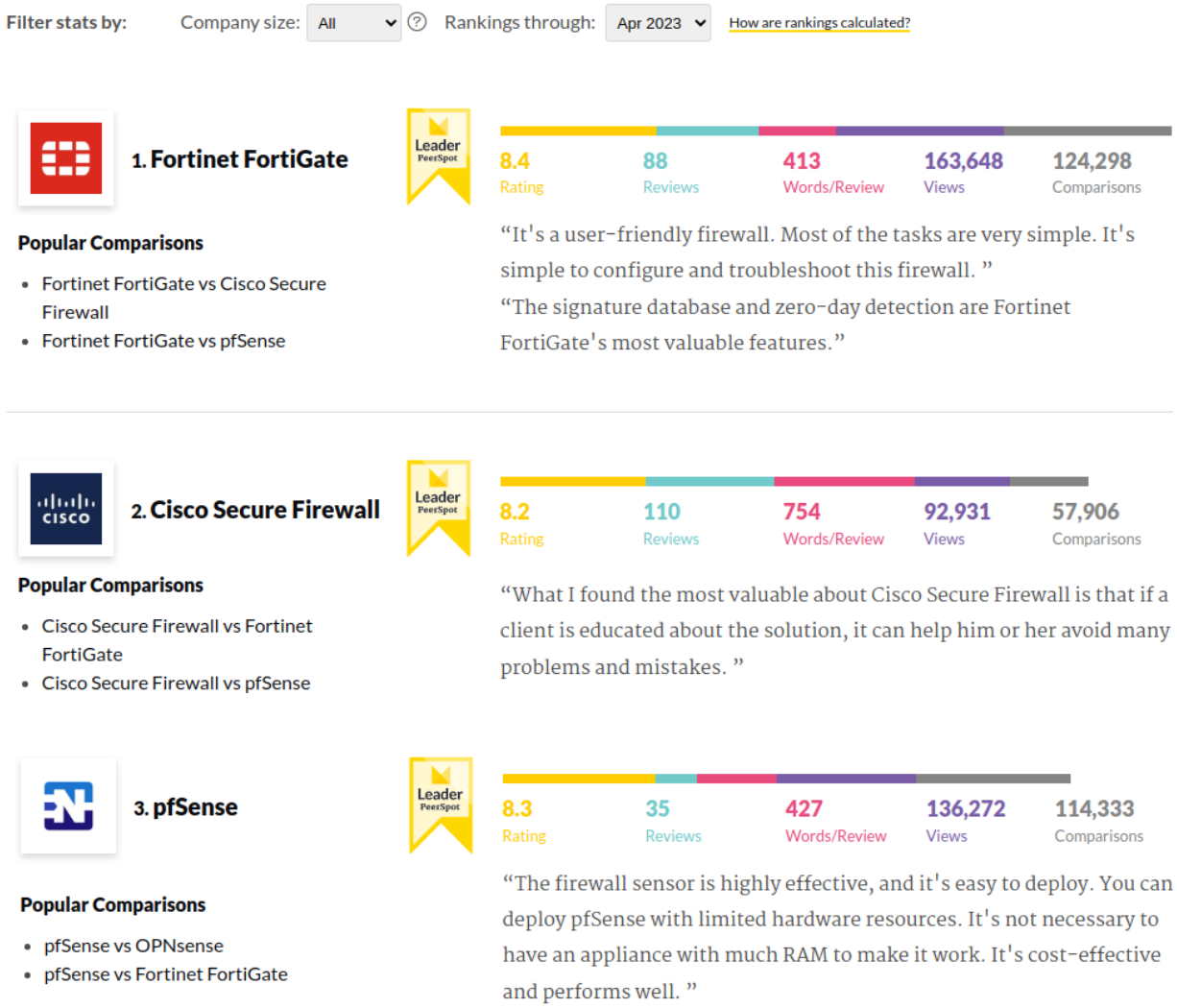


Figura 2.8: Recorte do ranqueamento de melhores soluções em *Firewall* apresentado pela plataforma PeerSport
Fonte: <https://www.peerspot.com/categories/firewalls>, Acessado em 10/05/2023

Com o pfSense não é necessário um equipamento específico, podendo ele ser instalado em

qualquer máquina ou equipamento físico, ou virtual que tenha os requisitos mínimos listados no site do sistema, além de ser uma solução gratuita.

2.7.6 Wireshark

Segundo (WIRESHARK-FOUNDATION, 2023) o Wireshark é um analisador de protocolos de rede que permite verificar o que ocorre na rede a um nível microscópico.

A aplicação captura o tráfego da placa de rede com ajuda de bibliotecas específicas, estas, atuam na camada de enlace registrando até mesmo quadros que seriam normalmente rejeitados pela placa, ou simplesmente não seriam repassados para as camadas superiores. Isso permite ao administrador ter uma boa visão do que está se passando na rede. Com a configuração correta da placa de rede e do enlace é possível visualizar praticamente todo o tráfego da rede. A aplicação permite salvar a captura para análise posterior, filtros que ajudam a isolar pacotes com determinadas características, gráficos e vários outros recursos que auxiliam na análise do tráfego. A aplicação é largamente utilizada pelo mundo em instituições de ensino e empresas de TIC. Treinamentos e certificações são oferecidos pela *Chappell University* antiga *Wireshark Univerty* em <*https://www.chappell-university.com/*>, a tela da ferramenta pode ser vista na figura 2.9.

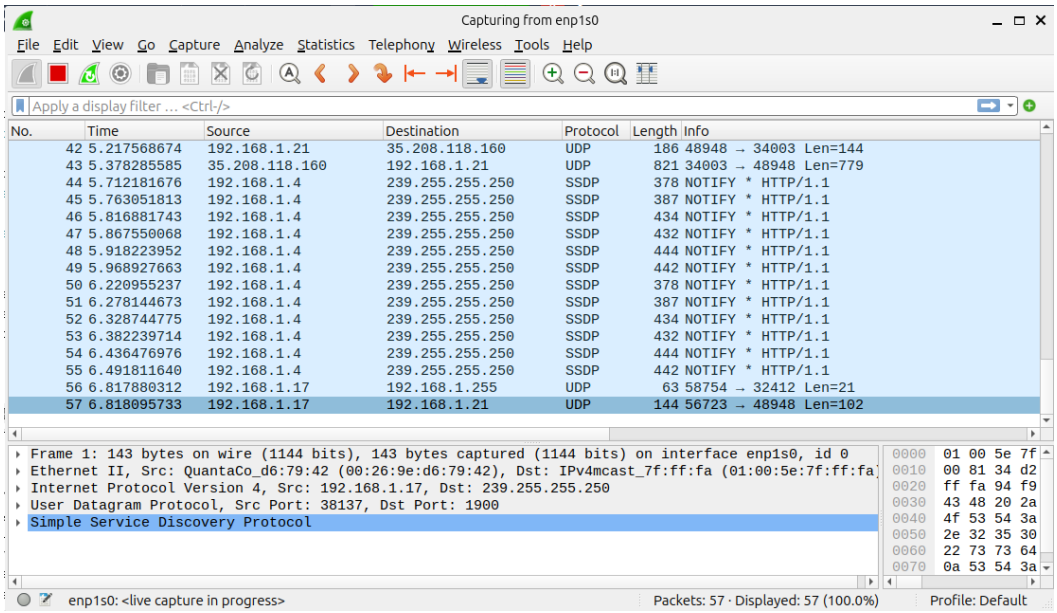


Figura 2.9: Wireshark

2.7.7 DD-WRT

Segundo (EMBEDD, 2023), O DD-WRT é um *firmware*, ou sistema operacional embarcado, para roteadores sem fio, construído sobre Linux. O sistema é largamente utilizado como alternativa ao *firmware* oficial de alguns fabricantes, substituindo o *firmware* original em alguns modelos é possível obter mais recursos e maior controle o roteador. Neste trabalho o DD-WRT foi virtu-

alizado para fazer o papel de roteador de uma rede local. A tela do *firmware* pode ser vista na figura 2.10.

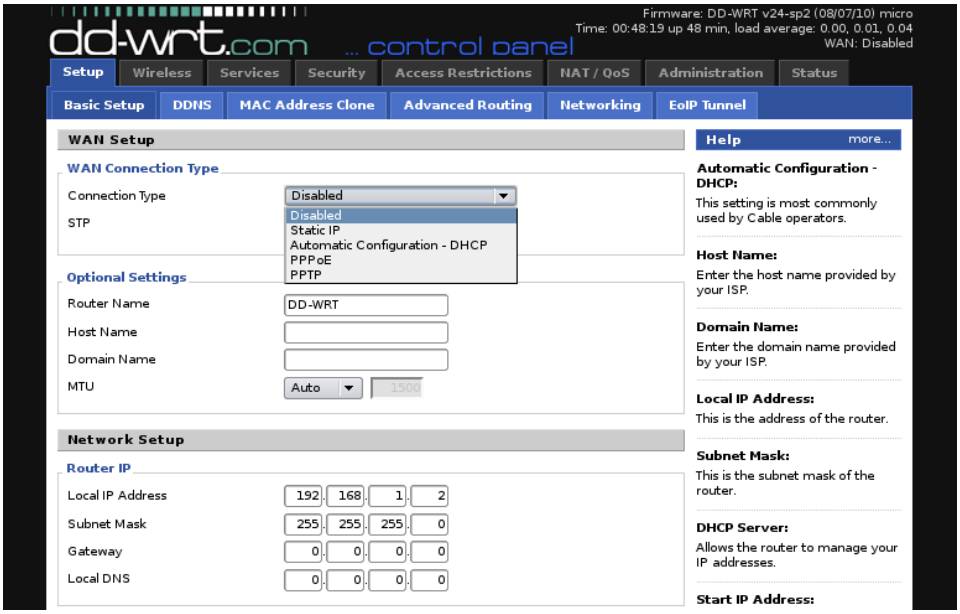


Figura 2.10: DD-WRT

2.7.8 Zabbix

Segundo (ZABBIX-SIA, 2023) o Zabbix é uma aplicação gratuita que monitora diversos parâmetros de estado de servidores, aplicações, serviços dentre outros elementos de rede. Permite a configuração de alerta baseados em eventos, possibilitando rápida reação a problemas. A aplicação possui diversos recursos, dentre eles a criação de mapas de infraestrutura que ajudam no monitoramento não só dos ativos e serviços de rede mas também do estado dos enlaces entre os ativos.

Existe no mercado uma grande variedade de ferramentas de monitoramento remoto, algumas pagas e outras gratuitas, dentre as ferramentas gratuitas o Zabbix é uma das mais bem avaliadas e das mais utilizadas. No site da Gartner é possível ver algumas comparações de ferramentas, a figura 2.11 traz uma comparação do Zabbix com outra ferramenta de monitoramento bem conhecida, o Nagios.

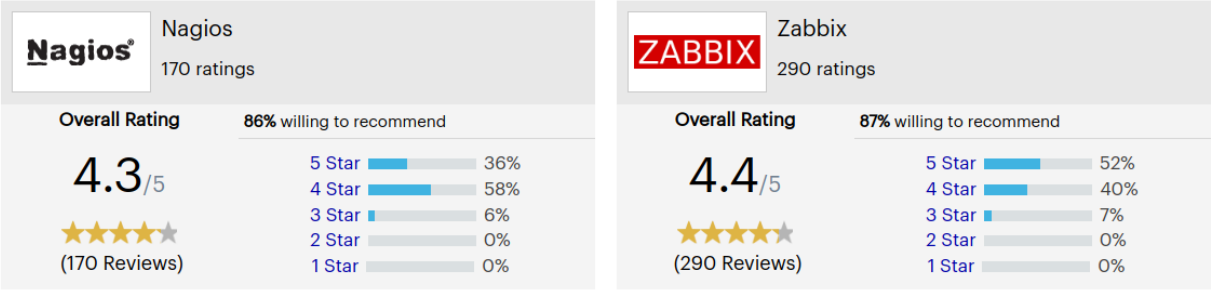


Figura 2.11: Comparação entre Zabbix e Nagios baseada em avaliações de usuários. Fonte: <https://www.gartner.com/reviews/market/infrastructure-monitoring-tools/compare/nagios-vs-zabbix>, Acessado em 25/05/2023

2.7.9 Wazuh

A aplicação Wazuh de acordo com seus desenvolvedores (WAZUH-INC, 2023), é uma plataforma de segurança de código aberto, com funcionalidades de SIEM capaz de oferecer segurança tanto em ambientes convencionais de TIC quanto em ambientes virtualizados, conteinerizados e em nuvem.

A plataforma pode ser utilizada para análise de logs, detecção de *rootkits*, detecção de vulnerabilidades, checagem de integridade de arquivos, inventário, dentre outros.

A plataforma possui componentes centrais e de *endpoits*, os componentes de *endpoits* são agentes de coleta instalados nos dispositivos finais, tais como, servidores, máquinas virtuais, estações de trabalhos e outros. Esses agentes coletam e enviam os dados para os componentes centrais. Os componentes centrais são: Wazuh server, Wazuh dashboard e Wazuh indexer. O W. server gerencia os agentes, recebe os dados coletados e envia para o W. indexer. O W. indexer funciona como indexador dos dados e como motor de busca. O W. dashborad funciona como interface *Web* para visualização e análise dos dados. A figura 2.12 mostra a arquitetura da plataforma.

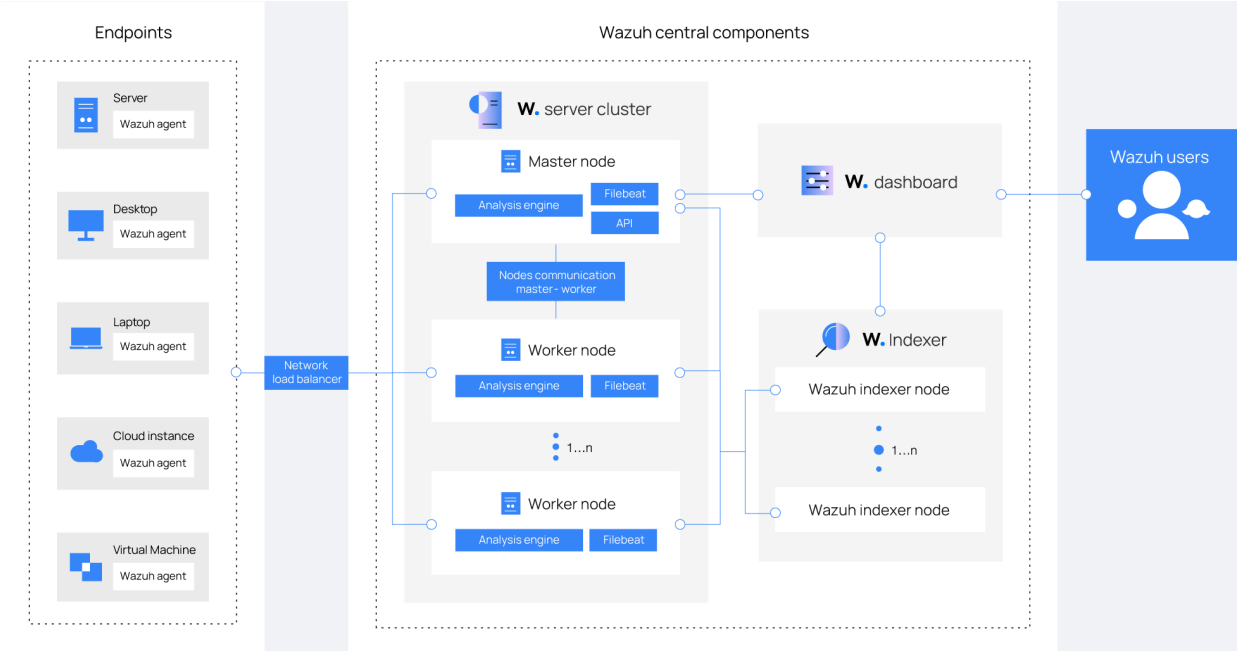


Figura 2.12: Arquitetura de componentes da plataforma Wazuh. Fonte: <https://documentation.wazuh.com/current/getting-started/architecture.html>, Acessado em 25/05/2023

2.7.10 ZeroTier

De acordo com (ZEROTIER-INC, 2023) o ZeroTier é um monitor de recursos virtuais de rede, que trabalha sobre uma rede ponto a ponto global criptograficamente segura. Fornecendo virtualização de rede avançada e recursos de gerenciamento equivalentes a um *switch* SDN em redes locais e de longa distâncias.

Com o ZeroTier é possível criar uma rede virtual e adicionar dispositivos reais que podem estar em qualquer parte do mundo com acesso à internet, esses dispositivos serão capazes de se comunicar através dessa rede.

Um aplicativo gerenciador de conexões é instalado nos dispositivos finais, permitindo que estes se conectem às redes virtuais criadas na infraestrutura do ZeroTier, deste modo, os dispositivos conectados na mesma rede pode se comunicar.

Conforme a documentação do desenvolvedor, trata-se de uma rede ponto a ponto, onde existem servidores principais que conhecem todos os nós. Quando um nó A deseja enviar informação para um nó B e ainda não conhece um caminho direto, A envia para um servidor raiz R que encaminha para B, ao mesmo tempo, R envia mensagens para A e B ensinando caminhos por onde os dois podem se comunicar sem passar por R. Quando A e B recebem suas mensagens, estabelecem uma conexão ponto a ponto direta entre eles e conseguem se comunicar de forma mais rápida.

Nesta arquitetura os nós raízes funcionam como serviço de localização de todos os nós. Após estabelecido o caminho direto entre os nós que querem se comunicar, as informações enviadas não passam mais por um ponto central como numa VPN convencional, o que resulta numa conexão

com menor latência.

2.8 TRABALHOS RELACIONADOS

Este trabalho visa estabelecer um esquema para o monitoramento e gerenciamento de remoto de redes, de modo que foram efetuadas buscas por referências em trabalhos sobre ferramentas de monitoramento remoto.

Em (GOSENHEIMER; NOGUEIRA, 2022) os autores fazem uma avaliação de algumas ferramentas de detecção de ataques, neste trabalho foi criada uma topologia de rede em ambiente virtual, onde foram implantadas três soluções de EDR e alguns ataques foram feitos para checar a capacidade de detecção e alerta das soluções. As soluções EDRs utilizadas foram: OSSEC+, OpenEDR e Wazuh, das quais o Wazuh foi a solução que detectou a maior quantidade de ataques. No presente trabalho serão testados cenários virtuais e físicos onde o Wazuh será implantado para estes de segurança.

Em (HOLANDA; SILVA, 2022) os autores propõem uma infraestrutura de rede virtualizada, porém, bem detalhada e abrangente, onde realizam a implementação de uma gama de aplicações para monitoramento e gerenciamento dos ativos de rede. No decorrer do trabalho é feita a análise da integração entre as ferramentas de monitoramento. Ao final do trabalho os autores concluem que a implantação e integração das soluções, comprem o objetivo de ser uma fonte importante de informações para auxiliar profissionais da área. No presente trabalho o conjunto de soluções de monitoramento, conexão e segurança será implantado e testado não apenas em ambiente virtual mas também em ambiente físico, trazendo para os testes variáveis do mundo real.

3 ESTRUTURA DA PROPOSTA DO ESQUEMA DE MONITORAMENTO

Neste capítulo são apresentados os cenários utilizados nos testes e o cenário final da proposta, uma discussão sobre os motivos de algumas escolhas e a demonstração de algumas configurações mais específicas.

Na figura 3.1 é possível visualizar as etapas percorridas até chegar à construção do esquema final da proposta.

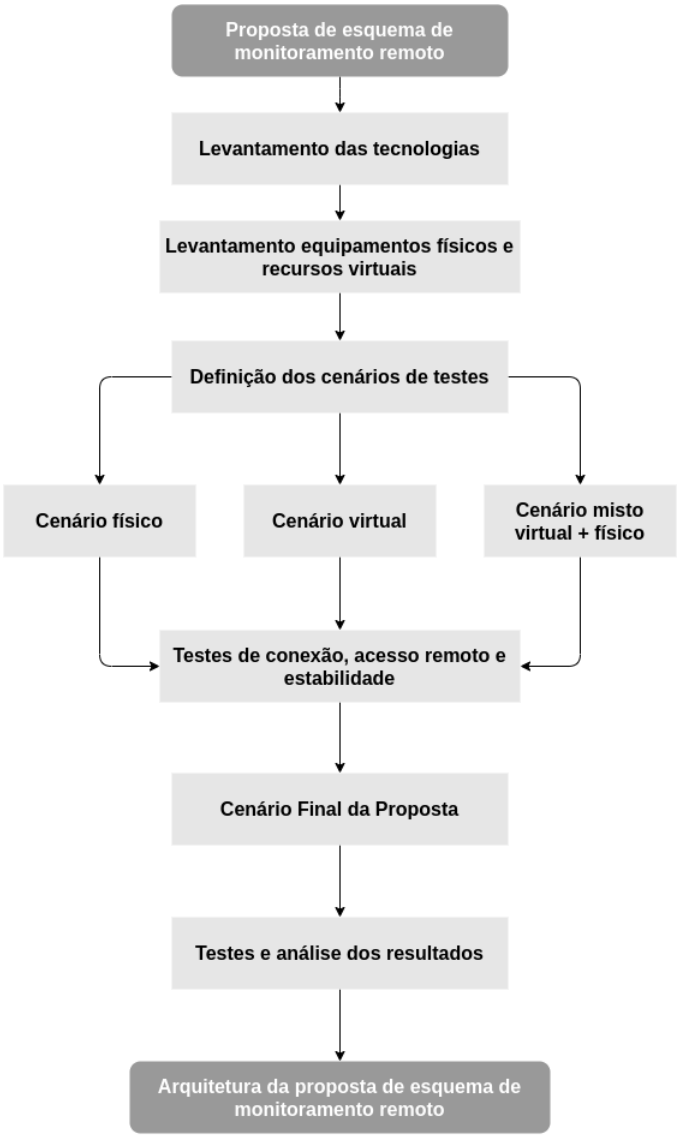


Figura 3.1: Etapas para construção do esquema da proposta.

Levantamento das Tecnologias - realização do levantamento dos *softwares* e das soluções

para conexão remota, monitoramento e segurança.

Levantamento dos equipamentos - levantamento dos equipamentos físicos disponíveis para montagem dos cenários físicos. Levantamento dos recursos virtais para criação dos cenários virtuais.

Definição dos cenários de testes - após levantamento dos recursos físicos e virtuais, inicia-se a etapa de definição dos cenários. Onde foram definidos cenários do tipo totalmente físico, totalmente virtual, e hibrido de virtual e físico.

Testes de conexão, acesso remoto e estabilidade - após a criação dos primeiros cenários, testes de conexão, acesso remoto e estabilidade da rede foram aplicados a estes cenários.

Cenário final - após os testes iniciais, o cenário final da proposta foi definido.

Testes e análise dos resultados - com o cenário final definido, testes aplicados anteriormente foram reaplicados aplicados, também foram aplicados novos testes, dessa vez na área de segurança e detecção de ataques. Por fim é realizada a apresentação do resumo dos resultados e a análise dos mesmos.

3.0.1 ZeroTier

A configuração padrão do ZeroTier começa pela criação de uma rede na interface *Web* da plataforma, na configuração da rede é possível configurar a faixa de endereçamento IP privada a ser utilizada na rede. Em seguida, instalar o gerenciador de conexão em cada máquina que fará parte da nova rede, e por meio deste, ingressar cada máquina utilizando o identificador único da rede. O ingresso de cada dispositivo dever ser autorizado via interface *Web*, quando cada máquina receberá um endereço IP na rede virtual e serão capazes de se comunicar entre si. A figura 3.2 mostra um esquema básico de conexão de vários dispositivos através do ZeroTier.

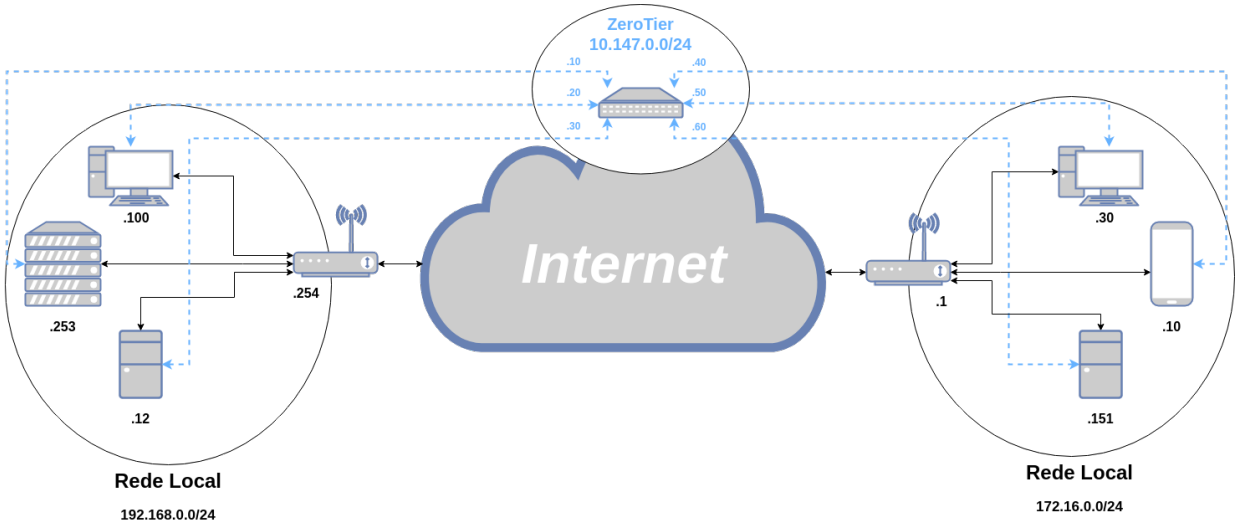


Figura 3.2: Esquema básico de conexão com ZeroTier.

Na configuração padrão vista anteriormente, sempre que um dispositivo novo é conectado a uma das redes físicas remotas, é necessário instalar e configurar o gerenciador de conexões do ZeroTier. A proposta deste trabalho é ingressar apenas um dispositivo por rede no Zerotier, e utilizar esse dispositivo como *gateway* para interconectar as redes físicas através da rede virtual, criando assim um esquema similar a uma VPN *Site-to-Site*, de modo que todas as máquinas nas duas redes possam se comunicar, e que qualquer novo dispositivo conectado a uma das redes físicas também tenha acesso às duas redes automaticamente. O dispositivo que funcionará como *gateway* da rede virtual, pode ser o próprio roteador de borda da rede, ou outro dispositivo inserido na rede, basta que tenha suporte ao gerenciador de conexão do ZeroTier. No caso da necessidade de controle de acesso em algum sentido entre as redes, um *firewall* pode ser utilizado.

A figura 3.3 mostra o esquema *site-to-site* com um *gateway* ZeroTier, nesta configuração, todo o tráfego gerado na rede 192.168.0.0/24 com destino à rede 172.16.0.0/24 será encaminhado para a interface ZeroTier no rotador R-A e todo tráfego da rede 172.16.0.0/24 com destino à rede 192.168.0.0/24 será encaminhado pelo roteador R-B de volta à máquina GW-Zerotier na mesma rede, e então encaminhado pela interface ZeroTier. Na interface *Web* do ZeroTier é possível criar uma regra de roteamento que permita que quadros com destino a rede 192.168.0.0/24 sejam encaminhados à máquina 10.147.0.30 e quadros com destino à rede 172.16.0.0/24 sejam encaminhados à máquina 10.147.0.60. Deste modo conseguimos prover conectividade entre qualquer máquina em ambas as redes.

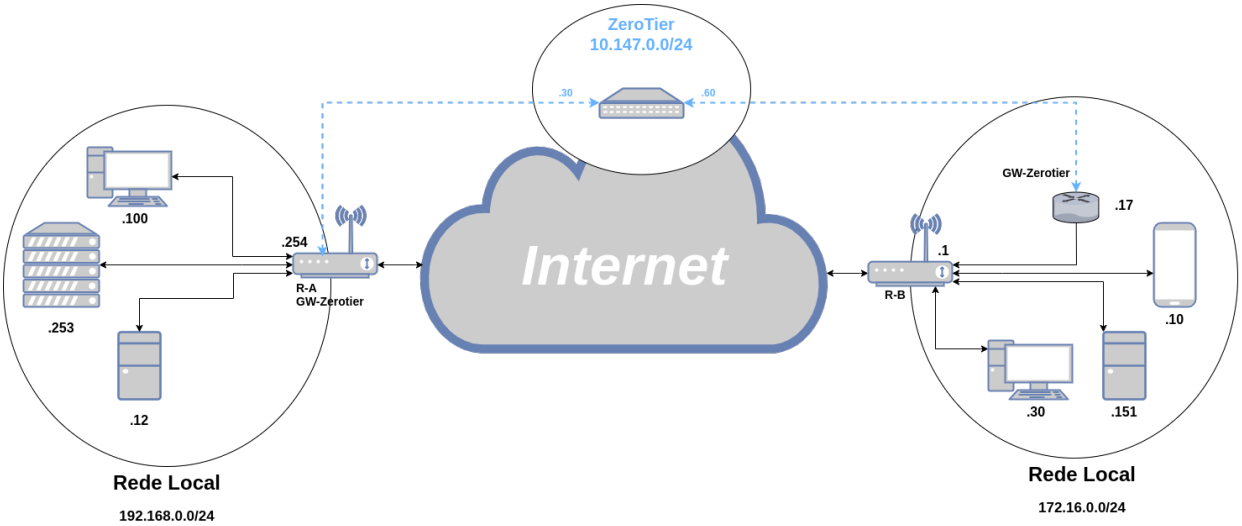


Figura 3.3: Esquema com *gateway* ZeroTier.

3.1 EQUIPAMENTO FÍSICO

Para a construção dos cenários e testes, foram utilizados alguns equipamentos físicos, a tabela 3.1 mostra cada equipamento e suas características, além do apelido dado ao equipamento para facilitar o entendimento na montagem dos cenários na fase de testes.

Apelido	Máquina	Processador	Memória	Disco
M1	*notebook* DELL Latitude 3400	Intel® Core™ i7-8565U × 8	32GB	M.2 nvme 512 GB
M2	*notebook* DELL Latitude 3400	Intel® Core™ i7-8565U × 8	16GB	M.2 nvme 512 GB
M3	*notebook* Lenovo ThinkPad E470 M3	Intel® Core™ i7-7500U × 4	8GB	500 GB
M4	*notebook* Avell M4	Intel® Core™ i7-7700HQ × 8	32GB	1.5TB
M5	*notebook* Lenovo IdeaPad 320	Intel® Core™ i7-7500U × 4	16GB	2TB
M6	*descktop mini PC* DELL OptiPlex 3050	Intel® Core™ i3-7100T × 4	16GB	480GB
M7	*notebook* Acer Aspire 1410	Genuine Intel @ CPU U2300 @ 1.20GHz x2 ×	3GB	120GB
R1	Roteador TP-Link TL-WR841ND 4 portas 10/100 Mbps	MIPS 535 MHz	32 MB	Flash 4MB

Tabela 3.1: Máquinas fisicas.

3.2 EQUIPAMENTOS VIRTUAIS E EMULADOR

Para a emulação e virtualização dos cenários, utilizamos o *software* GNS3, os sistemas operacionais utilizados foram virtualizados nos cenários utilizando o QEMU, onde é possível criar a máquina virtual completa e instalar o sistema operacional que irá se comportar como se estivesse em uma máquina física. Os sistemas operacionais utilizados podem ser vistos na tabela 3.2.

Aplicação	Sistema Operacional Base	Plataforma
DD-WRT v3.0	Linux Kernel 4.9.229	QEMU
Windows	Windows	QEMU
Web Sever	Ubuntu Server 20.04 LTS	QEMU
PFsense 2,6	FreeBSD 12,3	QEMU
Xubuntu	Ubuntu 18,04 LTS	QEMU
Zabbix 5	AlmaLinux 8,6	QEMU
Wazuh	Ubuntu Server 20.04 LTS	QEMU
Web Term	Debian 8	Docker
Kali Linux 2023.2	Debian	QEMU

Tabela 3.2: Sistemas Operacionais e aplicações.

Além das máquinas virtuais também existem alguns elementos virtualizados pelo próprio GNS3, como alguns tipos de *switch*, dentre os quais, utilizamos o *switch ethernet*, que funciona como um *switch* básico e com algumas opções de configuração.

O GNS3 ainda fornece elementos de conexão, que permitem conectar as redes virtualizadas

com redes externas e por consequência com a própria *Internet*, neste caso temos duas opções:

3.3 MÁQUINAS VIRTUAIS

No momento da criação das máquinas virtuais que vão hospedar os sistemas operacionais e aplicações, é necessário especificar os recursos que serão utilizados pela máquina virtualizada como: memória, disco rígido, processador, placas de rede e outros. A escolha da quantidade de recursos alocados para cada máquina, deve levar em consideração a necessidade de processamento que a aplicação hospedada necessita, a quantidade de recursos disponíveis na máquina física para compartilhamento e a quantidade de máquinas virtuais que irão funcionar ao mesmo tempo, compartilhando esses recursos. Também é importante ter em mente que a performance de determinadas aplicações está ligada à quantidade e qualidade dos recursos alocados para ela.

Na tabela 3.3 é possível ver a configuração padrão inicial criada para cada máquina virtual.

Máquina	**Memória**	**Disco Rígido**	**Processador**	**Interfaces de rede**
DD-WRT v3.0	1024 MB	170 MB	2	2
Windows	1024 MB	5,5 GB	1	1
Ubuntu Server	1024 MB	11 GB	1	1
PFsense 2,6	1024 MB	1,5 GB	1	2
Xubuntu	1024 MB	4,5 GB	1	1
Zabbix 5	2048 MB	1,5 GB	1	1
Wazuh	4096 MB	11 GB	3	1
Kali Linux 2023.2	1024 MB	13 GB	1	1

Tabela 3.3: Máquinas virtuais recursos.

3.4 CENÁRIOS

Nesta sessão, são apresentados os cenários utilizados, para que se tenha uma ideia da organização dos dispositivos na proposta, até chegar ao cenário final. Os testes realizados serão detalhados no capítulo de resultados.

3.4.1 Cenário - Topologia básica

O cenário da Topologia básica, visto na figura 3.4 serviu de base para os primeiros testes, feitos com a rede totalmente virtualizada e com parte da rede virtualizada e parte com máquinas reais. Mais detalhes, testes e resultados estão no capítulo de resultados.

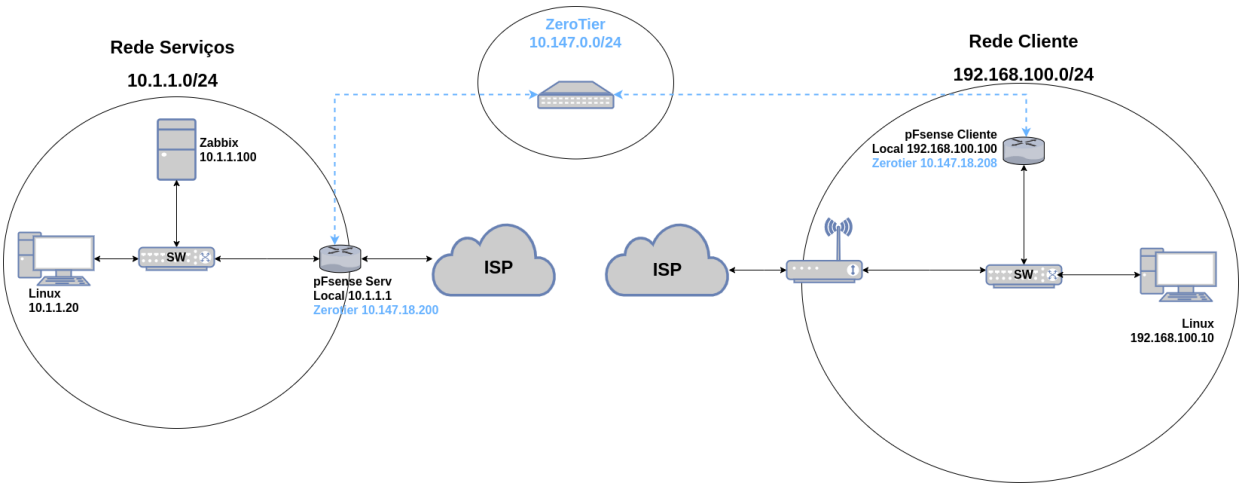


Figura 3.4: Topologia básica dos primeiros testes.

3.4.2 Cenário - Topologia maquinas físicas

O cenário visto na figura 3.5 foi utilizado para a realização dos testes com máquinas físicas, onde as duas redes são compostas por máquinas reais alocadas em locais geograficamente separados. Mais detalhes e testes no capítulo de resultados.

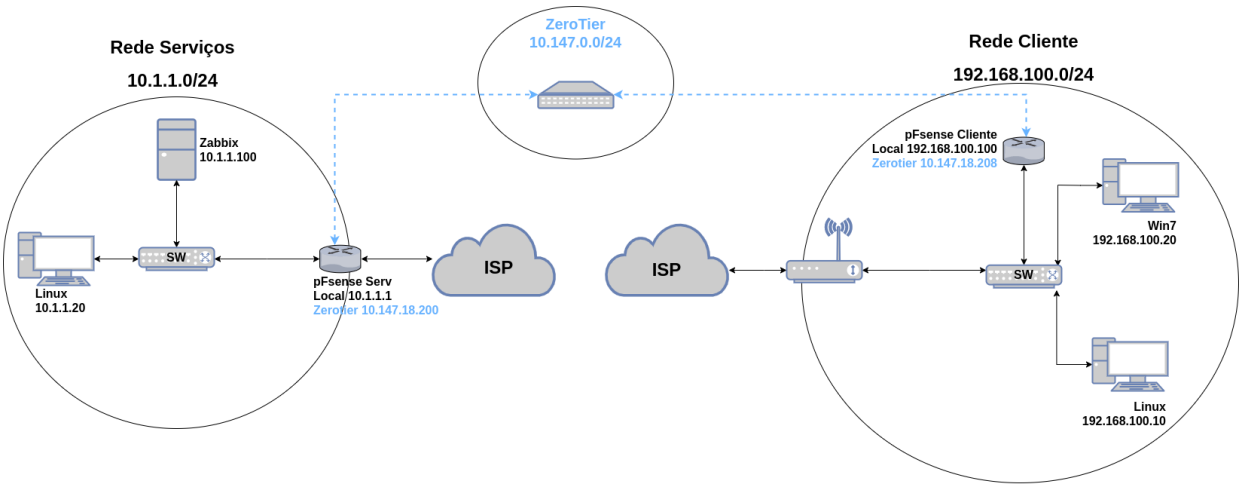


Figura 3.5: Topologia de testes com máquinas reais.

3.4.3 Cenário - Esquema Final

O cenário visto na figura 3.6 é representa o esquema final da proposta, e nele foram utilizadas todas a soluções descritas anteriormente, além da adição da máquina do atacante.

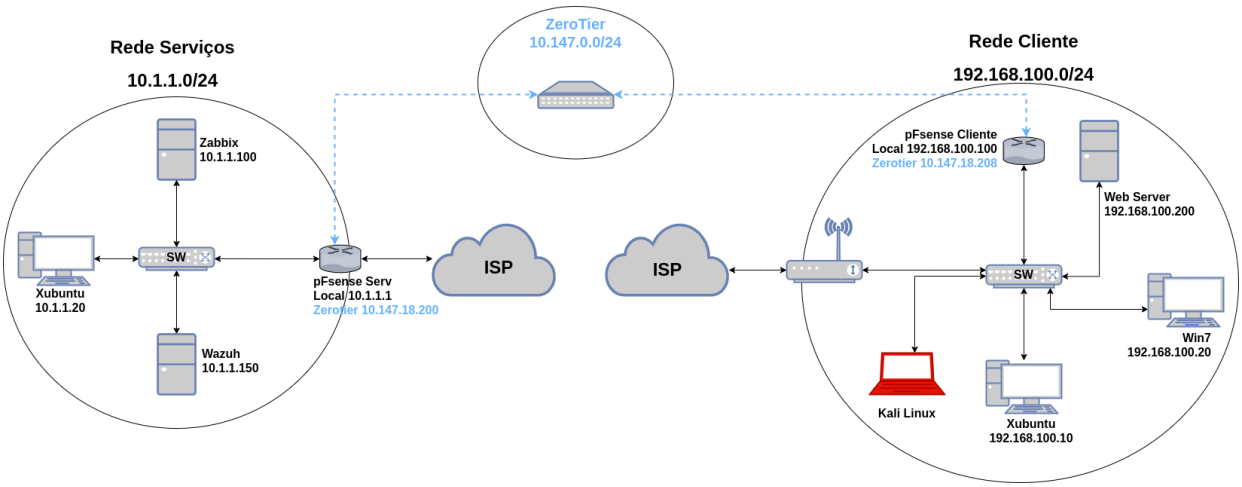


Figura 3.6: Topologia do cenário final de testes.

3.5 CONFIGURAÇÕES

Nesta sessão apresentamos as configurações realizadas nas máquinas virtuais e nas aplicações. Algumas configurações mais básicas como, instalação de sistemas operacionais e algumas aplicações, configurações de redes e outras, não foram documentadas neste trabalho, por sua natureza básica e por estarem vastamente documentadas na *Internet*. Outras aplicações utilizadas neste trabalho, necessitam de configurações específicas para que funcionem do modo como esperamos, estas serão descritas com maiores detalhes.

As configurações demonstradas aqui foram utilizadas principalmente no cenário final do trabalho, porém, a maioria também se aplica aos outros cenários apresentados, em alguns casos com pequenas adaptações.

3.5.1 Configuração da rede virtual ZeroTier

Para utilizar o ZeroTier é necessário criar um usuário no *site* da solução. Após criado o usuário já é liberado o acesso aos recursos do plano *Basic*, que é o plano gratuito, com ele é possível ter 1 administrador, até 25 dispositivos na mesma rede e quantidade de redes ilimitada. Existem dois outros planos pagos que liberam alguns recursos como, quantidades maiores de administradores e de dispositivos na mesma rede.

Ao acessar, o usuário é encaminhado para uma espécie de painel de controle, onde pode criar e gerenciar suas redes. Ao selecionar uma rede é possível editar as opções da rede e adicionar dispositivos nela, além de visualizar o identificador da rede, que deve ser usado nos dispositivos que vão ingressar nela.

Na figura 3.7 é possível visualizar a lista dos dispositivos conectados à rede virtual, com destaque para as duas máquinas pFsense utilizadas como *Gateway* ZeroTier no cenário final de

testes.

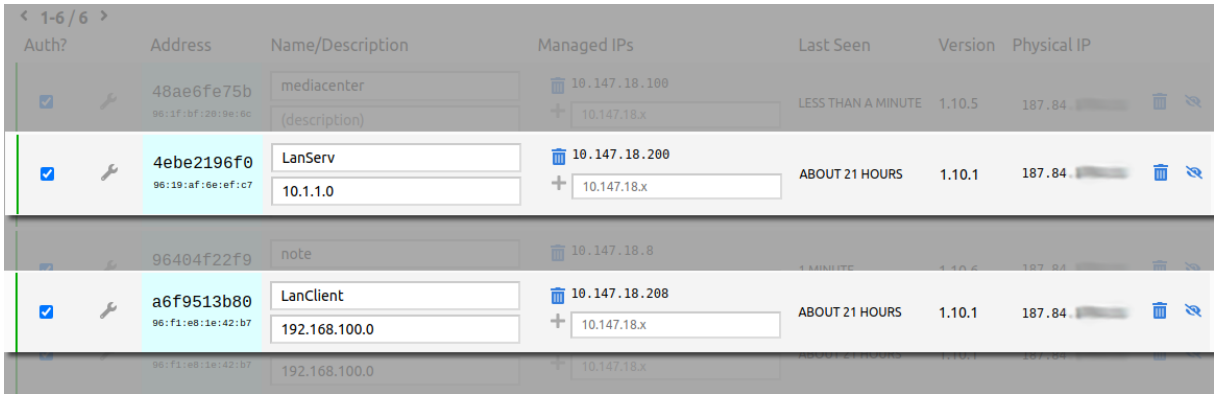


Figura 3.7: Lista de dispositivos conectados à rede virtual.

Ainda nas configurações da rede é possível criar regras de roteamento que serão injetadas nos dispositivos ingressados na rede. A figura 3.8 mostra a seção de rotas gerenciadas. Foi utilizado este recurso neste experimento para configurar o roteamento entre as redes "Rede de serviços"e "Rede cliente"passando pela rede virtual do ZeroTier no cenário final de testes.

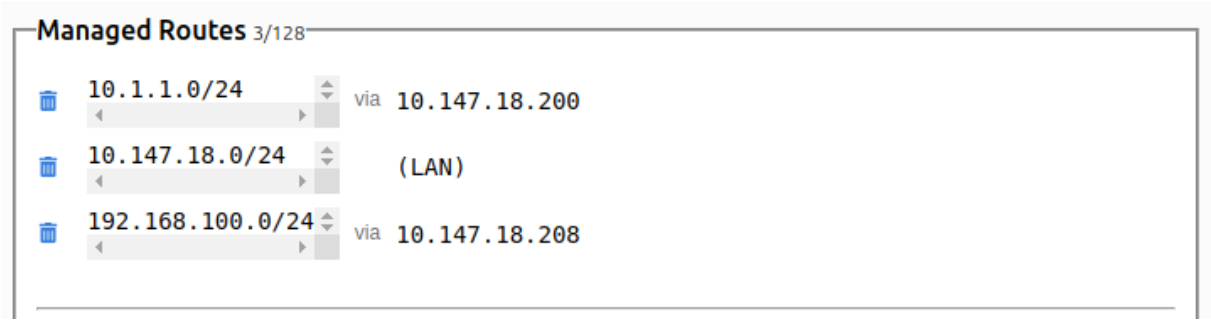


Figura 3.8: Gerenciador de Rotas na rede ZeroTier.

- A primeira regra encaminha quadros com destino à rede 10.1.1.0/24 para o dispositivo com endereço 10.147.18.200.
- A segunda regra encaminha quadros com destino à rede 10.147.18.0/24 para a rede local (rede virtualizada do ZeroTier).
- A terceira regra encaminha quadros com destino à rede 192.168.100.0/24 para o dispositivo com endereço 10.147.18.208.

3.5.2 Máquinas na rede Cliente

Configuração das máquinas instaladas na rede Cliente, a rede onde estão as máquinas que estão sendo monitoradas.

DD-WRT

Um *firmware* alternativo normalmente utilizado em roteadores sem fio. Antes de instalar é necessário verificar o fabricante, modelo e versão de *hardware*, pois para melhor aproveitar o *hardware* e os recursos de cada dispositivo, existem versões específicas do *firmware* para cada dispositivo suportado. Neste trabalho virtualizamos o *firmware* para funcionar como roteador, para isso usamos a versão para processadores x86 que atualmente não recebe mais suporte do projeto, porém continua disponível no *site* para *download*.

Instalação

- A instalação do *firmware* em ambiente virtual não é trivial e requer alguns passos mais específicos, por isso o processo é detalhado nos apêndices deste trabalho subseção 6.1.1.

Configuração

- A máquina é utilizada no cenário como roteador de borda de uma rede local, por tanto necessita de no mínimo duas interfaces de rede, sendo que na interface conectada a rede externa o endereçamento é via DHCP externo, e na interface da rede local o endereçamento é fixo e o dispositivo fornece DHCP para a rede local.

- Para a configuração com *gateway* ZeroTier estilo VPN *site-to-site* funcionar, é necessário que os quadros saindo da rede local, com destino à rede remota, sejam encaminhados de volta para máquina com ZeroTier, para isso foi configurada uma rota estática no DD-WRT.

- Como o roteador está na borda da rede é necessário ativar a tradução NAT para que as máquinas internas consigam acessar redes externas como a *Internet*.

A figura 3.9 mostra alguns detalhes da configuração.

- O detalhe A da figura 3.9 mostra os campos de configuração de rota estática.
- O detalhe B da figura 3.9 destaca duas rotas da tabela:
  - Uma rota *default* que aponta para um endereço externo.
  - Uma rota estática específica que encaminha pacotes destinados à rede 10.0.0.0/8 para o endereço 192.168.100.100 na rede local.
- O detalhe C da figura 3.9 mostra a configuração do NAT para tradução dos quadros saindo pela interface eth0 com destino a redes externas.

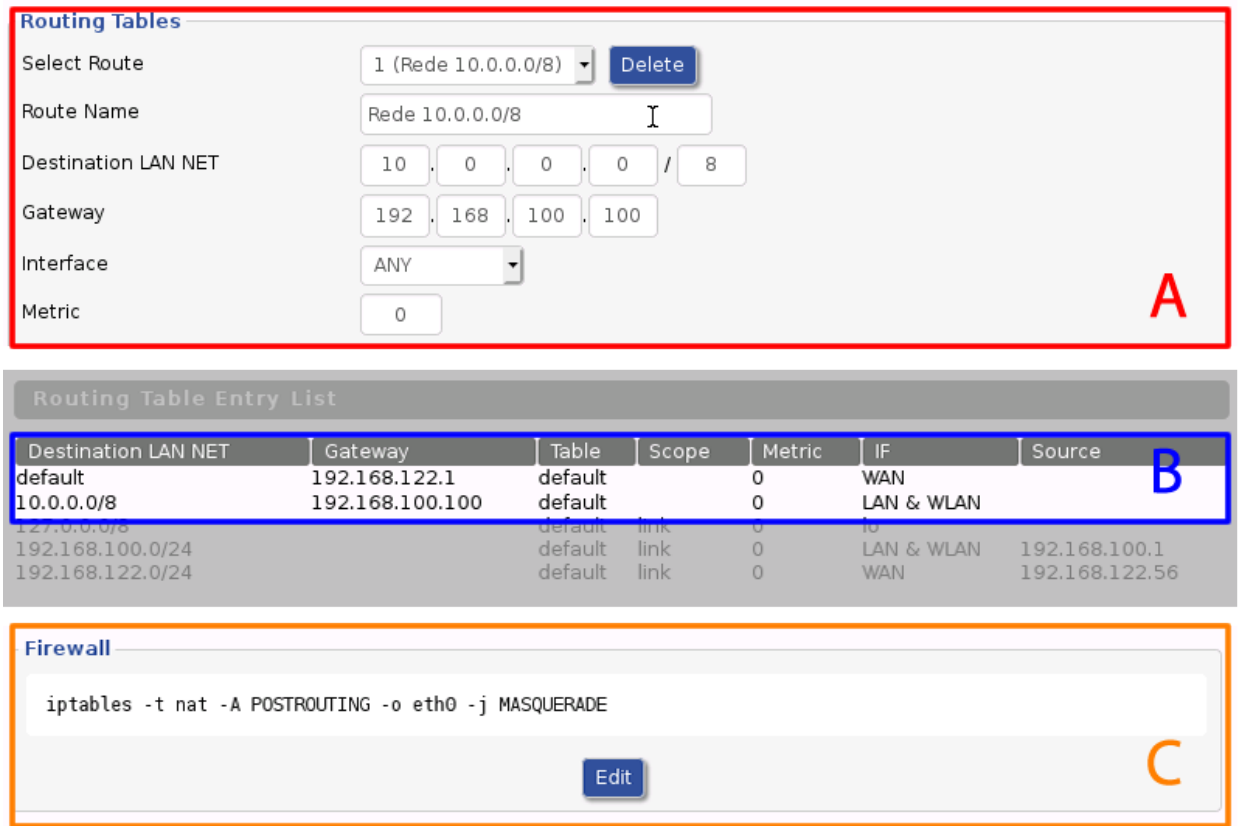


Figura 3.9: Detalhes da configuração de rotas e NAT do DD-WRT

pFsense

Como dito na subseção 2.7.5, o pFsense é um **software** voltado para redes, construído sobre uma base FreeBSD. Sua instalação é relativamente simples, basta acessar o *site* do projeto, baixar a imagem de instalação, selecionando a arquitetura de processador correspondente à máquina onde será instalado e seguir com a instalação. A configuração inicial pode ser feita no terminal da própria máquina estilo *command line interface* CLI, ou, pode-se conectar uma máquina com interface gráfica na interface *local area network* LAN do pFsense e acessar a interface de gerência *Web*.

O pFsense possui uma grande variedade de pacotes extras de recursos que podem ser instalados através de seu próprio gerenciador de pacotes, porém, ainda não possui um pacote oficial com o gerenciador de conexões do ZeroTier. Existe, no entanto, uma versão criada pela comunidade FreeBSD para a versão 12 do sistema, que pode ser instalada.

O processo de preparar, compilar e instalar o módulo é longo e requer alguns passos específicos, por isso preferimos detalhar o processo nos apêndices deste trabalho subseção 6.1.2. Nesta seção são demonstradas as configurações mais simples, feitas na interface *Web* do pFsense.

ZeroTier no pfSense - após a instalação do gerenciador de conexão do ZeroTier é necessário configurar para ingressar na rede virtual criada. As figuras 3.10 e 3.11 mostram a ativação e configuração do ZeroTier.

Configurações na figura 3.10:

- Figura 3.10 detalhe A - Após a instalação do módulo, é possível acessar a opção ZeroTier VPN.
- Figura 3.10 detalhe B - Marcar a opção Enable em VPN/Zerotier/Configuration.
- Figura 3.10 detalhe C - Em VPN/Zerotier/Networks adicionar o identificador da rede virtual criada para ingressar a máquina.
- Figura 3.10 detalhe D - Para permitir que quadros sejam encaminhados através do túnel na rede virtual é necessária uma configuração no sistema. Em System/Advanced/System Tunables/Edit adicionar Tunable = net.link.tap.up_on_open, Value= 1.
- Figura 3.10 detalhe E - Após habilitado o ZeroTier cria uma interface de rede virtual na máquina, agora é necessário atribuir essa interface à lista de interfaces do pFsense.
- Figura 3.10 detalhe F - Depois de ter sido reconhecida pelo sistema é necessário habilitar a interface. Não é necessário configurar outros parâmetros como endereçamento, pois a interface receberá essas informações do ZeroTier.

Configurações na figura 3.11:

- Figura 3.11 detalhe G - É necessário criar as regras de *firewall* que permitam o tráfego na nova interface, neste caso optamos por liberar a passagem de qualquer pacote.
- Figura 3.11 detalhe H - Voltando ao menu VPN/Zerotier/Networks é possível verificar o estado da conexão com a rede virtual, neste caso a conexão ocorreu com sucesso.
- Figura 3.11 detalhe I - No memu de estado das tabelas de rotas do pFsense é possível ver uma rota da rede 10.1.1.0/24 apontando para o endereço 10.147.18.200 que é a inteface ZeroTier da máquina pFsense da rede de serviços de monitoramento, e saindo pela interface viritual local iniciada com zt, essa rota foi configurada na interface *Web* do ZeroTier e injetada na máquina por ele, assim como uma configuração SDN seria.

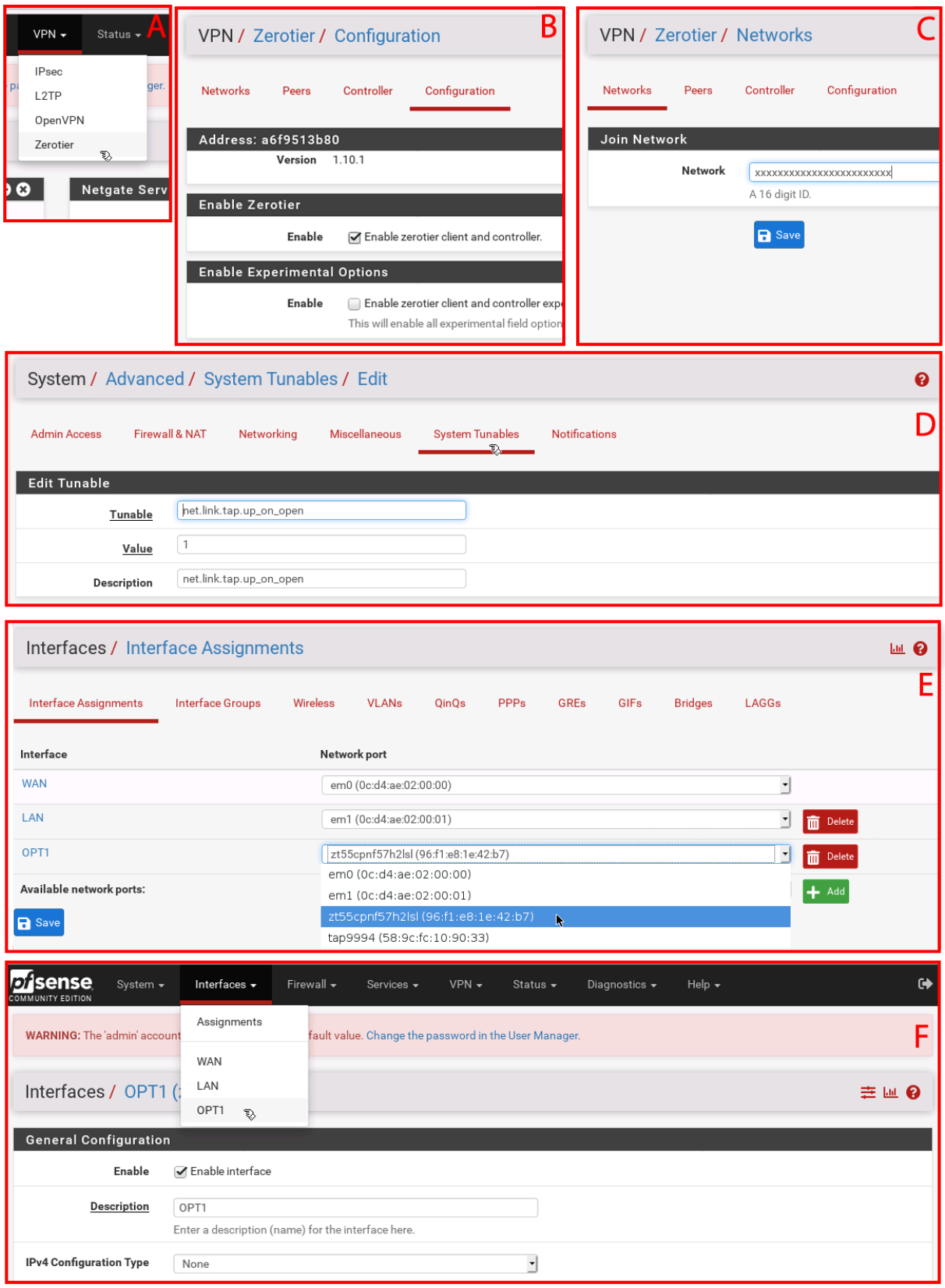


Figura 3.10: Detalhes da configuração ZeroTier no pFsense parte 01

Firewall / Rules / OPT1

Floating LAN OPT1

G

Rules (Drag to Change Order)

	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
✓	0 /0 B	IPv4 *	*	*	*	*	*	none			

VPN / Zerotier / Networks

Networks Peers Controller Configuration

H

Zerotier Networks

Status	Network	Type	Addresses	Interface	Bridged	Actions
OK	52b337794f115795 205	PRIVATE	10.147.18.208/24	OPT1 zt55cpnf57h2lsl	No	

Join

IPv4 Routes

I

Destination	Gateway	Flags	Uses	MTU	Interface	Expire
default	192.168.100.1	UGS	6548	1500	em1	
10.1.1.0/24	10.147.18.200	UGS	0	2800	zt55cpnf57h2lsl	
10.147.18.0/24	link#7	U	0	2800	zt55cpnf57h2lsl	
10.147.18.208	link#7	UHS	0	16384	lo0	
127.0.0.1	link#4	UH	194	16384	lo0	
192.168.100.0/24	link#2	U	4933	1500	em1	
192.168.100.100	link#2	UHS	0	16384	lo0	

Figura 3.11: Detalhes da configuração ZeroTier no pFsense parte 02

pFsense - Zabbix *agent* - o monitoramento do Zabbix pode ser realizado via protocolo *Internet Control Message Protocol* ICMP, *Simple Network Management Protocol* SNMP ou via agente instalado na máquina a ser monitorada, neste trabalho foi implementado o agente local devido à facilidade de configuração e estabilidade na conexão cliente-servidor, o pFsense tem um módulo chamado Zabbix-agent que pode ser instalado através do gerenciador de pacotes do pFsense, como pode ser visto na figura 3.12.

A figura 3.12 mostra o pacote Zabbix-agent instalado e a configuração.

- Figura 3.12 detalhe A - gerenciador de pacotes do pFsense mostra o pacote Zabbix-agent instalado.
- Figura 3.12 detalhe B - na configuração do agente é necessário informar o endereço IP do servidor Zabbix, que neste caso está em outra rede sendo acessado através da rede virtual do ZeroTier.

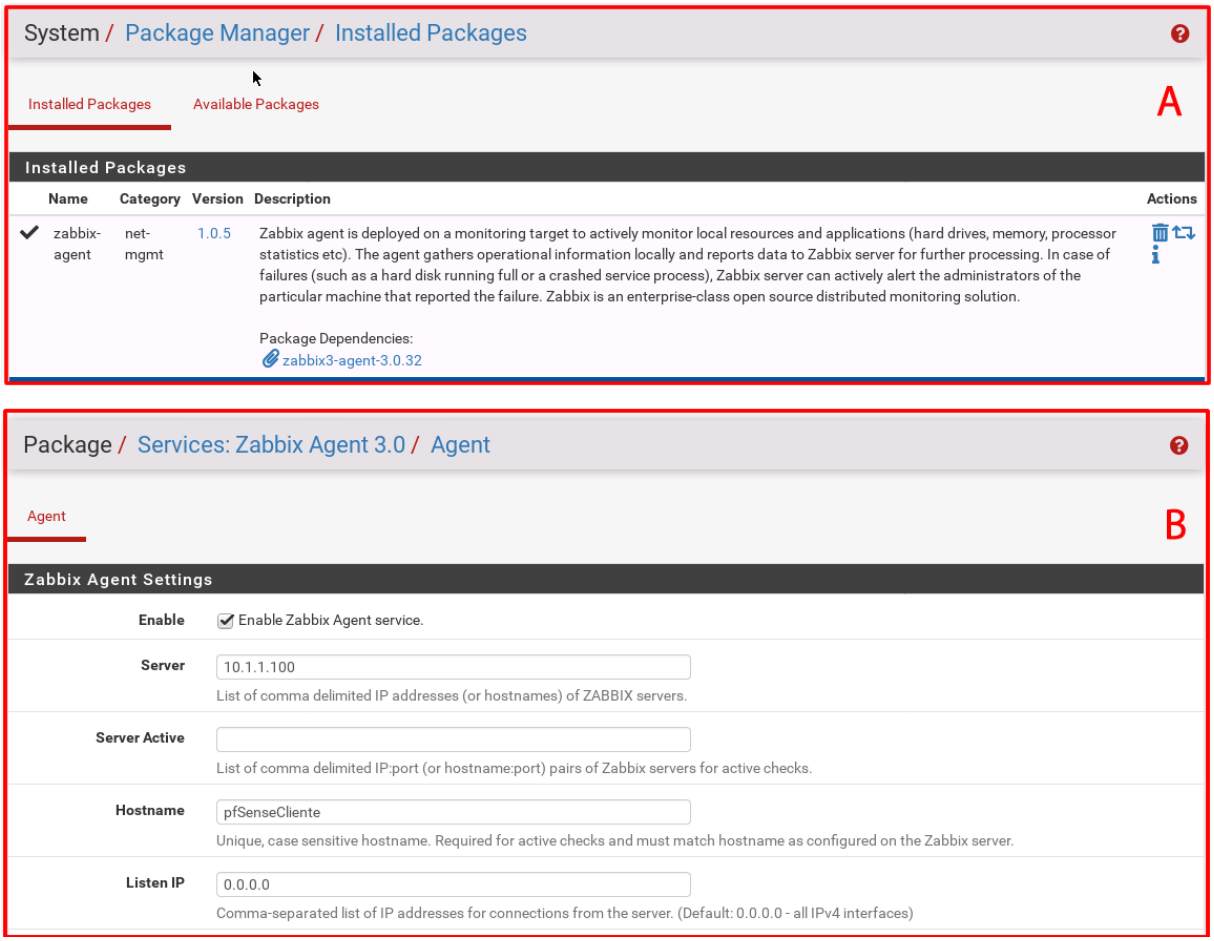


Figura 3.12: Pacote Zabbix-agent no pFsense

pFsense - Wazuh *agent* - para realizar a coleta de dados e possibilitar o monitoramento da máquina pela aplicação Wazuh é necessário instalar o agente de coleta de dados. O pFsense não possui pacote do agente em seu repositório oficial, porém é possível instalar o pacote do repositório do FreeBSD. O pacote disponível no para FreeBSD deve ser compilado na máquina antes da instalação, este é um processo longo, e requer alguns passos específicos, estando descrito nos apêndices desse trabalho subseção 6.1.3. Nesta seção são demonstradas as configurações do agente já instalado.

Configuração do agente após a instalação:

```
#Copiar /etc/locatime para /var/ossec/etc
cp /etc/localtime /var/ossec/etc

#Copiar /var/ossec/etc/ossec.conf.sample para ossec.conf e editar para a sua configuracao.
cp /var/ossec/etc/ossec.conf.sample /var/ossec/etc/ossec.conf
nano /var/ossec/etc/ossec.conf
```

```
#procurar pela linha <server><address>IP</address></server> e adicionar o endereco IP do servidor
<server>
    <address>10.1.1.150<address>
</server>

#adicionar wazuh agent ao /etc/rc.conf
nano /etc/rc.conf
sysrc wazuh_agent_enable="YES"

#Habilitar o servico
service wazuh-agent enable

#Iniciar o servico
service wazuh-agent start
```

Linux Server

Web server* Zabbix e Wazuh *agents - Uma máquina do tipo servidor, foi inserida nos cenários, para representar algum tipo de servidor interno presente na rede, como um servidor de domínio, servidor de impressão ou mesmo servidor de arquivos, neste caso, utilizamos um servidor *Web*, que também será monitorado pelas aplicações na rede de serviços. O sistema operacional utilizado foi Ubuntu Server 20.04 LTS. Foram instalados o Apache como serviço *Web*, e os agentes Zabbix e Wazuh para realizar a coleta de dados e verificação do estado da máquina. A listagem de comandos abaixo demonstra a instalação foi realizada.

```
#Instalacao do servidor Web
apt udate
apt install apache2

#=====================================================================
#Instalacao Zabbix agent
apt update
apt install zabbix-agent

#Editar o arquivo de configuracao
#Procurar as linhas "Server=127.0.0.1" e "ServerActive=127.0.0.1" e colocar o endereco IP do servidor remoto
nano /etc/zabbix/zabbix_agent.conf
Server=10.1.1.100
ServerActive=10.1.1.100

#Reiniciar o servico zabbix-agent
systemctl restart zabbix-agent

#Verificar o status do servico zabbix-agent
systemctl restart zabbix-agent
```

```
#=====================================================================
#Instalacao do Wazuh agent
#Baixar o instalador pra a versao de sistema operacional correta
curl -so wazuh-agent-4.3.10.deb https://packages.wazuh.com/4.x/apt/pool/main/w/wazuh-agent/wazuh-agent_4.3.10-1_amd64.deb

#Instalar e apontar o endereco do servidor remoto
sudo WAZUH_MANAGER='10.1.1.150' dpkg -i ./wazuh-agent-4.3.10.deb.

# Habilitar o agente como servico e iniciar o servico e verificar o estado.
systemctl daemon-reload
systemctl enable wazuh-agent
systemctl start wazuh-agent
systemctl status wazuh-agent

#O arquivo de configuracao fica em
/var/ossec/etc/ossec.conf
```

Linux *Descktop*

Linux *Descktop* Zabbix e Wazuh *agents* - uma máquina virtual Linux *Descktop* foi adicionada ao cenário para representar possíveis estações de trabalho Linux que podem ser encontradas em redes de clientes. Neste caso, a distribuição utilizada foi o Xubuntu, baseada na distribuição Ubuntu, porém com uma interface gráfica bem mais simples e leve, que, portanto exige menos recursos da máquina hospedeira da virtualização. Por ser uma distribuição baseada em Ubuntu, o Xubunto possui o mesmo sistema de arquivos e de serviços que o Ubuntu Server, assim sendo, os mesmos passos demostrados anteriormente para instalação e configuração dos agentes Zabbix e Wazuh podem ser aplicados ao Xubuntu.

Windows *Desktop*

Windows *Descktop* Zabbix e Wazuh *agents* - uma máquina Windows foi adicionada ao cenário para representar estações de trabalho Windows em redes clientes. A versão do Windows escolhida foi o Windows 7 devido ao fato de este ocupar menos espaço no disco rígido e funcionar bem com pouca memória e processamento, ou seja, exige menos recursos que as versões mais recentes do sistema. Assim como nos demais dispositivos, neste foram instalados os agentes de coleta e monitoramento do Zabbix e Wazuh.

```
#Instalacao do Zabbix agent
#Baixar o instalador do site:
#https://www.zabbix.com/br/download_agents
#Executar o instalador
#Durante a instalacao sera solicitado o endereco do servidor Zabbix
#Apos a instalacao o Zabbix Agent aparecera no gerenciador de servicos do Windows
```

```
7    #O arquivo de configuracao fica em: C:\Program Files\Zabbix Agent\
     zabbix_agentd.conf

9    #=====================================================================
     #Instalacao do Wazuh agent
11   #Para instalacao do Wazuh agente no Windows e necessario:
     #Ser administrador da maquina
13   #Ter PowerShell 3.0 ou superior
     #Executar os comandos abaixo no PowerShell como administrador, informando o
     endereco IP do servidor Wazuh
15
     Invoke-WebRequest -Uri https://packages.wazuh.com/4.x/windows/wazuh-agent
     -4.3.10-1.msi -OutFile env:tmp-agent-4.3.10.msi;msiexec.exe/i{env:tmp}\wazuh-agent
     -4.3.10.msi /q WAZUH_MANAGER='10.1.1.150' WAZUH_REGISTRATION_SERVER='10.1.1.150
     '
17
     #Apos a instalacao o Wazuh Agent aparecera no gerenciador de servicos do
     Windows
19   #O arquivo de configuracao fica em
     C:\Program Files\ossec-agent\ossec
```

3.5.3 Máquinas na rede de serviços de monitoramento

pfSense

O pFsense foi utilizado na rede de serviços como roteador de borda da rede, neste caso foram criadas 2 interfaces de rede, uma para a rede interna LAN, que recebeu endereçamento IP estático e manual, e outra para conexão coma a rede externa WAN, que recebe endereçamento do DHCP do elemento *Cloud* de GNS3, além da interface virtual. A instalação e configuração do gerenciador de conexões do ZeroTier é semelhante à já demostrada na subseção 3.5.2, a interface de rede virtual criada pelo ZeroTier recebe automaticamente o endereço IP e a rota que foram configurados na interface *Web* do ZeroTier, como demonstrado na subseção 3.5.1. Assim na configuração do pFsense na rede cliente, os testes foram realizados sem restrições no *firewall*. Os detalhes da configuração podem ser vistos na figura 3.13.

A figura 3.13 mostra alguns detalhes da configuração.

- O detalhe A da figura 3.13 mostra o estado das interfaces de rede, interface WAN na rede externa, interface LAN na rede interna e a interface virtual ZeroTier em sua rede virtual.
- O detalhe B da figura 3.13 mostra a tabela de rotas do sistema, com destaque para a rede 10.147.18.0/24 que aponta para a interface virtual do ZeroTier, e a rota para a rede 192.168.100.0/24 (rede cliente) que aponta para a interface com endereço IP 10.147.18.208 que é a interface da máquina pfsense remota da rede cliente na rede ZeroTier, esta rota foi criada na interface *Web* do ZeroTier e injetada na máquina.

As *flags* UGS significam:

- U - A rota está ativa *up*.
- G - Esta rota é um *gateway* que encaminha quadros para outras redes.
- S - Esta rota foi configurada estaticamente. Neste caso a rota foi injetada pelo recurso SDN do ZeroTier.

- O detalhe C da figura 3.13 mostra as regras de *firewall* para a interface ZeroTier no pFsense da rede de serviços.

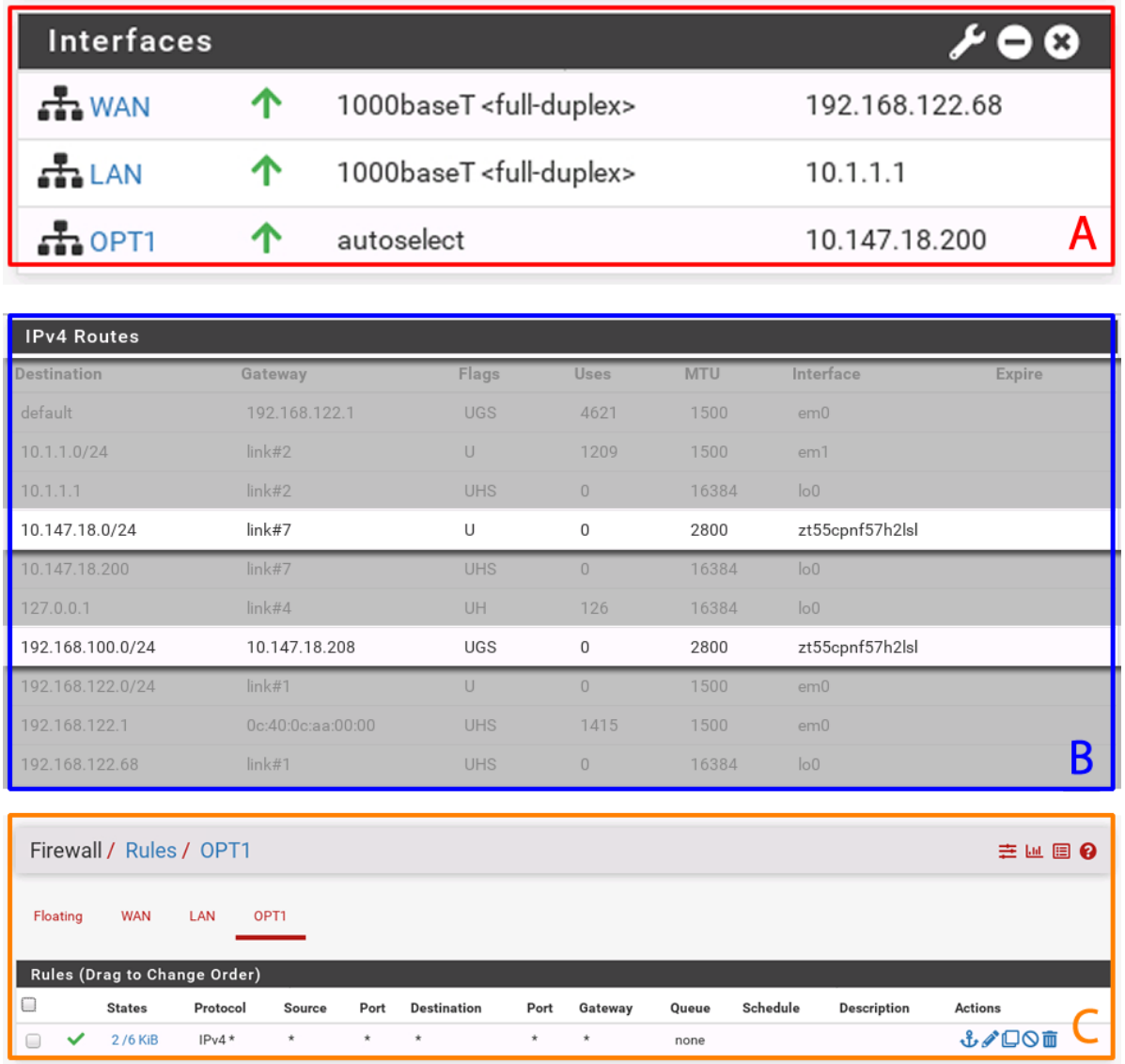


Figura 3.13: Detalhes da configuração pFsense na rede de serviços de monitoramento.

Zabbix

Como dito anteriormente na subseção 2.7.8 o Zabbix é uma aplicação para monitoramento de serviços e dispositivos, neste trabalho o sistema foi posicionado na rede de serviços de monitoramento e utilizado para monitorar os equipamentos na rede cliente. A solução é construída

sobre uma base Linux, no site da aplicação, na parte de *download* é possível escolher a distribuição Linux de base e o tipo de instalação que se deseja utilizar, via pacotes, nuvem ou imagem pronta. Neste trabalho foi utilizada a instalação via imagem. Após a instalação basta acessar a máquina e atribuir endereçamento IP, depois disso já é possível acessar a interface *Web*, adicionar os dispositivos a serem monitorados e criar o mapa da rede para ajudar na visualização.

Adicionar máquina para monitoramento

```
Configuration > Hosts > Create Host >
```

Aba Templates
Nesta aba é preciso escolher o *template* de monitoramento, que vem com gráficos dos itens monitorados na máquina. Posteriormente é possível adicionar outros gráficos e itens de monitoramento.

Aba Host
Nesta aba é preciso preencher os campos com informações da máquina a ser monitorada.

Após adicionar as máquinas é possível ver o estado destas na seção *Monitoring*.

```
1   Monitoring > Hosts >
```

A figura 3.14 mostra a seção de máquinas monitoradas, nela é possível ver o estado da máquina, se existem problemas e a quantidade de gráficos gerados para a máquina.

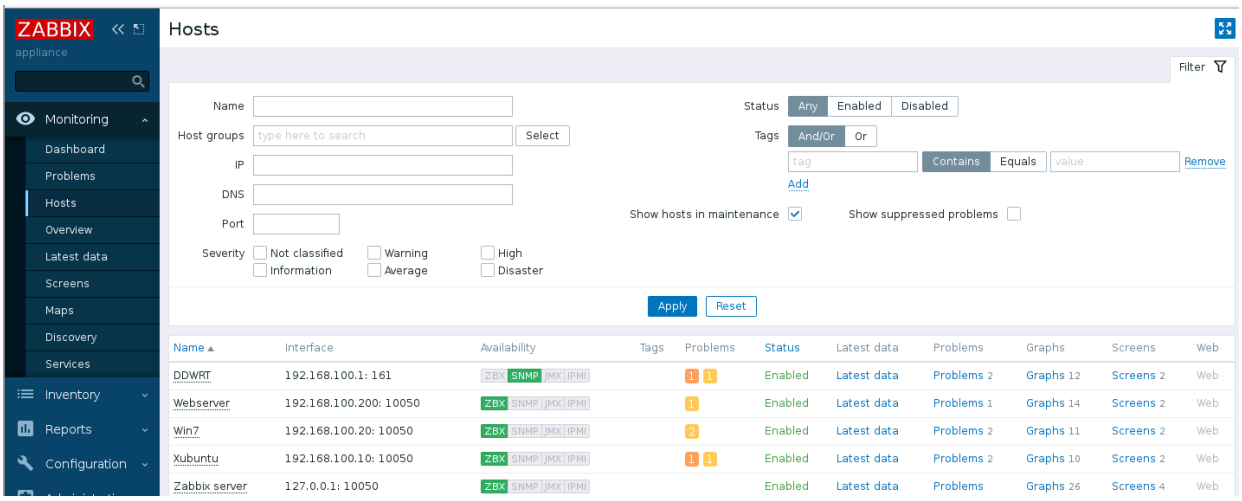


Figura 3.14: Máquinas monitoradas pelo Zabbix.

Ao clicar em um item de uma das colunas, detalhes serão apresentados, por exemplo ao clicar no item *Graphs* da máquina Xubuntu e navegar até o gráfico do tráfego de rede é possível observar

o consumo de dados da máquina.

As figuras 3.15 e 3.16 mostram o monitoramento da interface de rede e da utilização do processador da máquina Xubuntu durante a execução de um vídeo com alta qualidade no site *YouTube*, gerando alto nível de tráfego de rede e utilização do processador.

A figura 3.15 mostra os gráficos de tráfego de rede e de utilização do processador durante a execução do vídeo.

- Gráfico de tráfego na interface de rede, mostra o tráfego de dados ultrapassando 24Mbps.
- Gráfico de utilização do processador, mostra a utilização batendo em 100%.

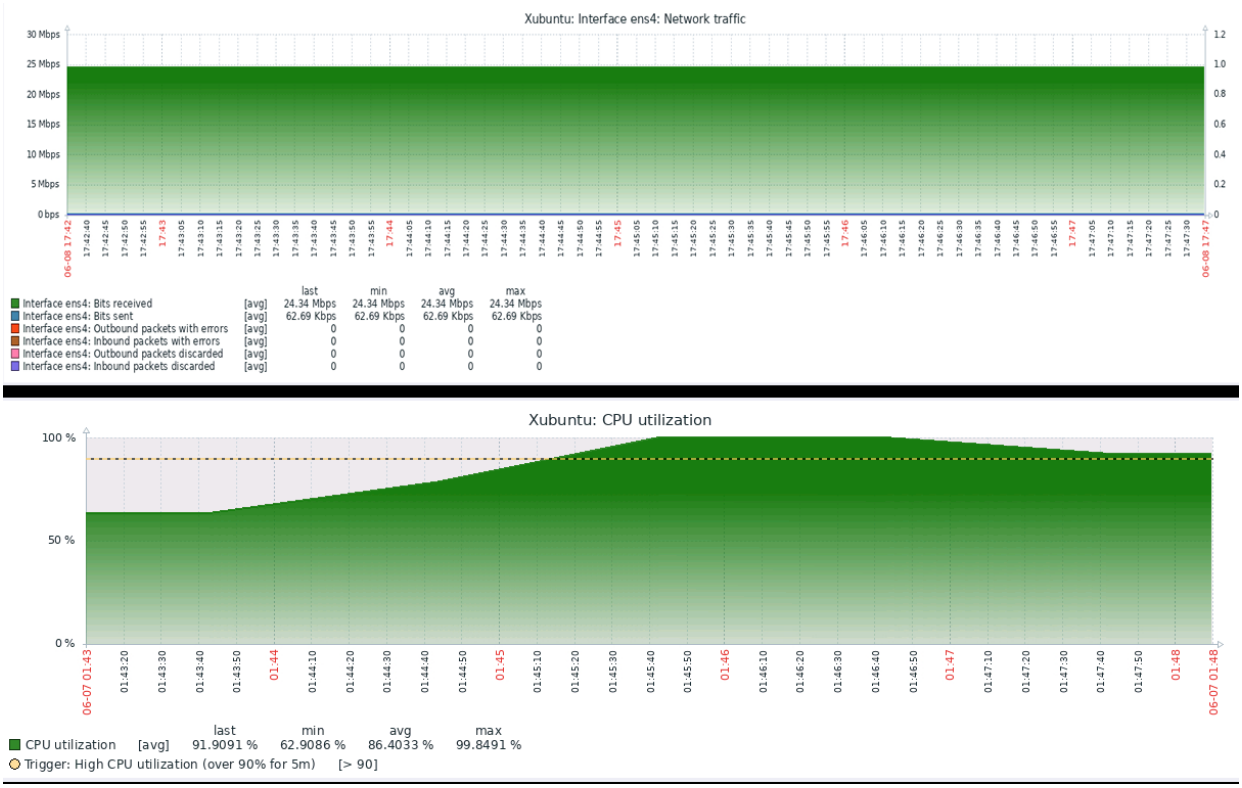


Figura 3.15: Gráficos da máquina Xubuntu durante execução de um vídeo.

A figura 3.16 mostra os gráficos de tráfego de rede e de utilização do processador após parar o vídeo e fechar o navegador.

- Gráfico de tráfego na interface de rede, mostra o tráfego de dados voltando para menos de 1Mbps.
- Gráfico de utilização do processador, mostra a utilização voltando para a média de 1%.

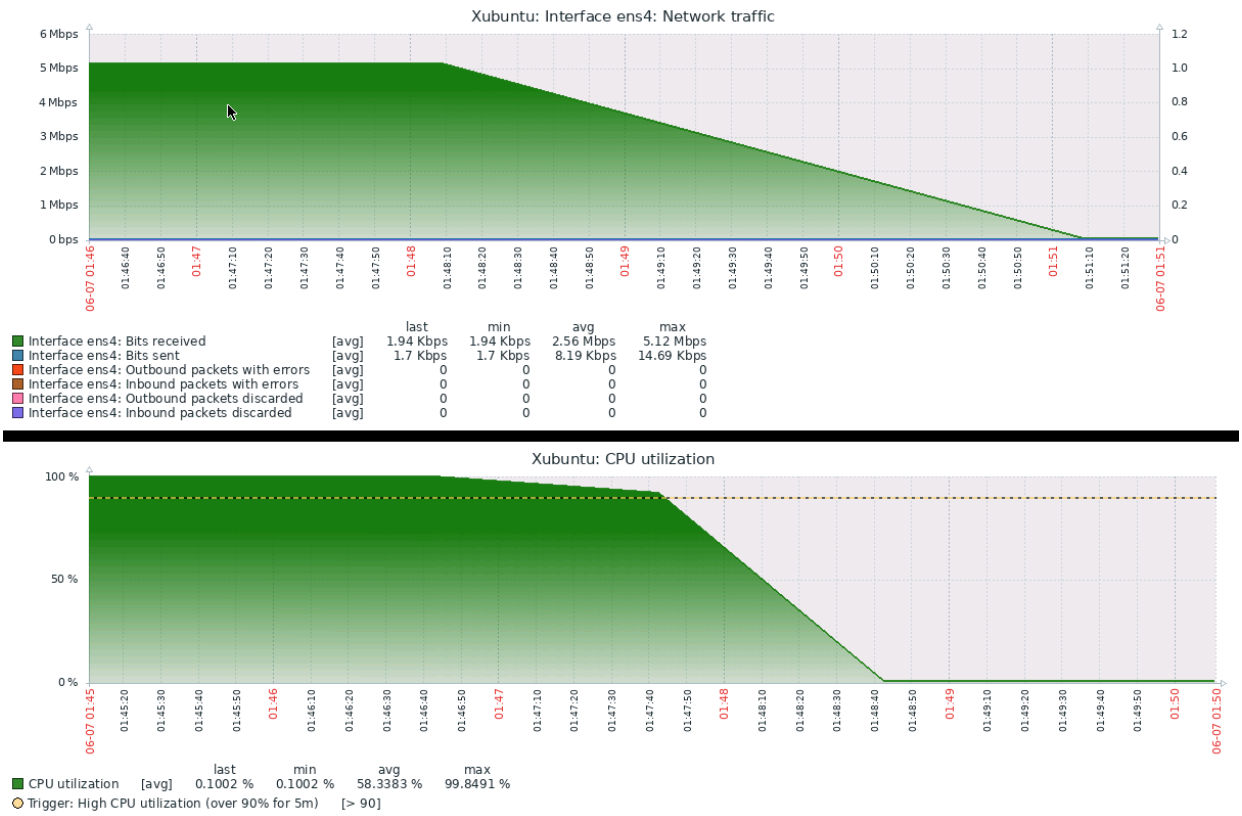


Figura 3.16: Gráficos da máquina Xubuntu depois da execução de um vídeo.

Além dos gráficos o Zabbix também permite a criação de mapas interativos, onde é possível acompanhar, por exemplo, os alertas de problemas e o tráfego de rede das máquinas.

A figura 3.17 mostra um mapa da rede, e nos permite visualizar o estado de cada máquina, os problemas detectados e o tráfego de rede. Os itens em destaque na imagem mostram o alto tráfego, ultrapassando 20Mbps, gerado em quanto a máquina Xubuntu na rede cliente consumia um vídeo em alta resolução vindo da *Internet*. É possível acompanhar o tráfego chegando pela interface externa do roteador DD-WRT e saindo pela interface interna, passando pelo *switch* e em seguida entrando pela interface do Xubuntu.

O processo para criação e configuração de mapas é extenso e requer muitos passos, porém é bem documentado na página da aplicação, essa documentação pode ser encontrada em: <https://www.zabbix.com/documentation/current/pt/manual/web_interface/frontend_sections/monitoring/maps>.

Outros gráficos também podem ser vistos na interface *Web* do Zabbix, assim como mais informações podem ser adicionadas aos mapas, esses recursos podem ser de grande valor para os administradores de rede, e podem auxiliar na tomada de decisão.

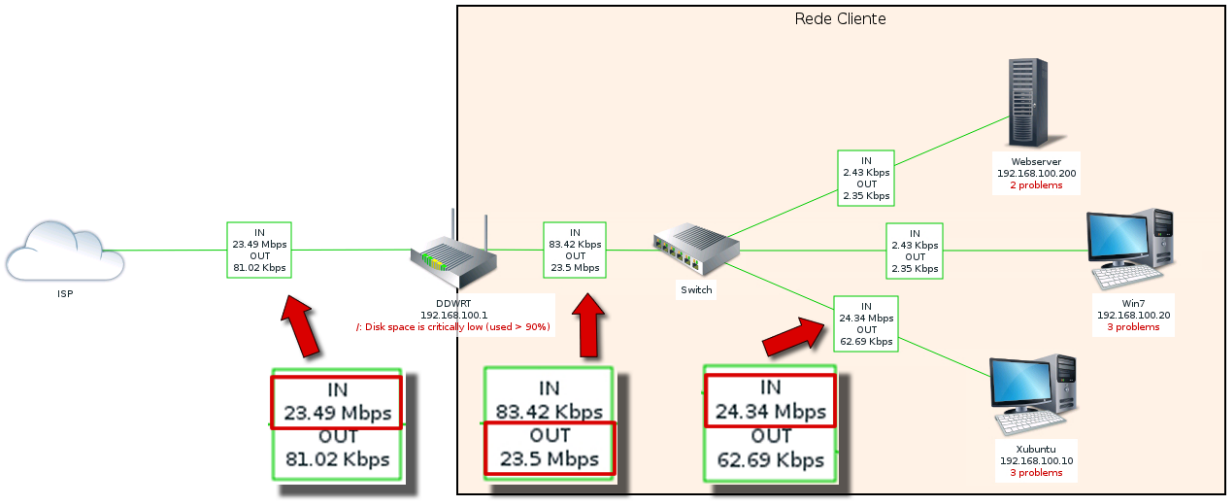


Figura 3.17: Mapa interativo da rede criado no Zabbix.

Ainda no Zabbix também é possível criar alertas que podem ser enviados aos administradores da rede de diversas formas, dentre elas, por *e-mail* e através de mensageiros eletrônicos como Telegram. A figura 3.18 mostra alguns alertas enviados pelo Telegram. Neste caso é possível criar disparos de alertas para grupos de usuários no Telegram.

- Figura 3.18 - detalhe A - mostra dois alertas de problema, indicando que as máquinas Xubuntu e Win7 estão indisponíveis. Em seguida as máquinas retornam e o sistema notifica os problemas foram resolvidos.
- Figura 3.18 - detalhe B - mostra dois alertas de problema, indicando que os serviços *Windows Defender* e Central de Segurança não estão ativos na máquina Win7. Em seguida os serviços se ativam e o sistema notifica que os problemas foram resolvidos.

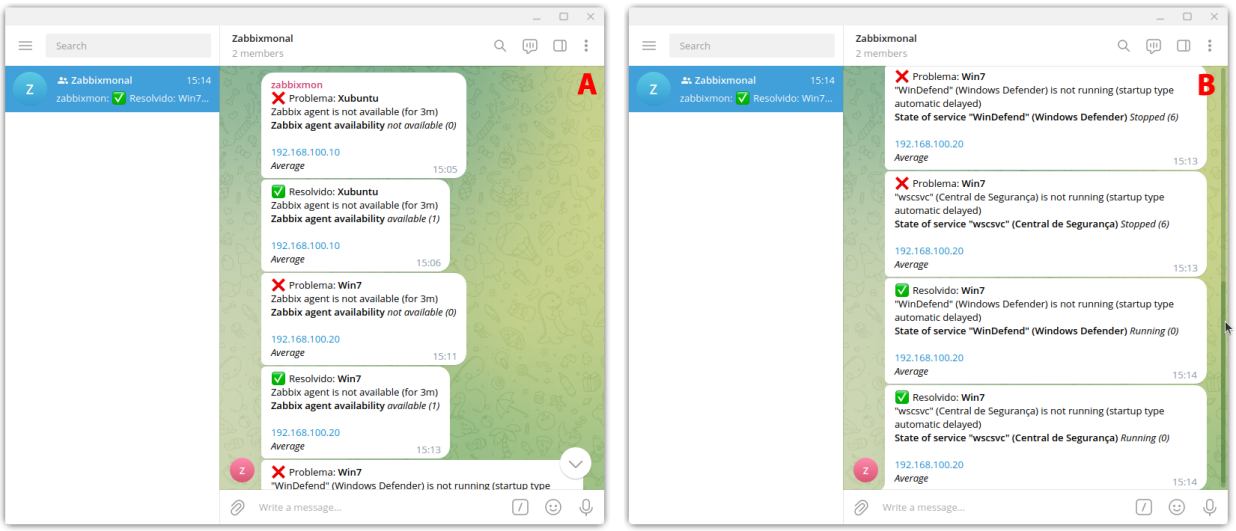


Figura 3.18: Alerta do Zabbix via Telegram.

Wazuh

Os componentes de gerenciamento do Wazuh foram instalados em uma máquina com Ubuntu *Server*, na rede de serviços de monitoramento seguindo as especificações da documentação, em seguida foram adicionados os agentes já instados e configurados nas máquinas clientes responsáveis pela coleta e envio dos dados ao servidor. A 3.19 mostra a tela de agentes conectados e a tela de eventos de segurança de uma das máquinas monitoradas.

- figura 3.19 - detalhe A - mostra a lista de máquinas monitoradas, onde é possível ver o estado da máquina e algumas informações básicas fornecidas pelos agentes instalados.
- figura 3.19 - detalhe B - mostra a tela do terminal da máquina Xubuntu, onde foi propositalmente realizada uma tentativa de elevação de privilégio com a senha errada por três vezes. Logo abaixo observamos a tela de ventos de segurança do Wazuh para o Xubuntu, onde pode se ver que o erro da senha gerou um evento de segurança.

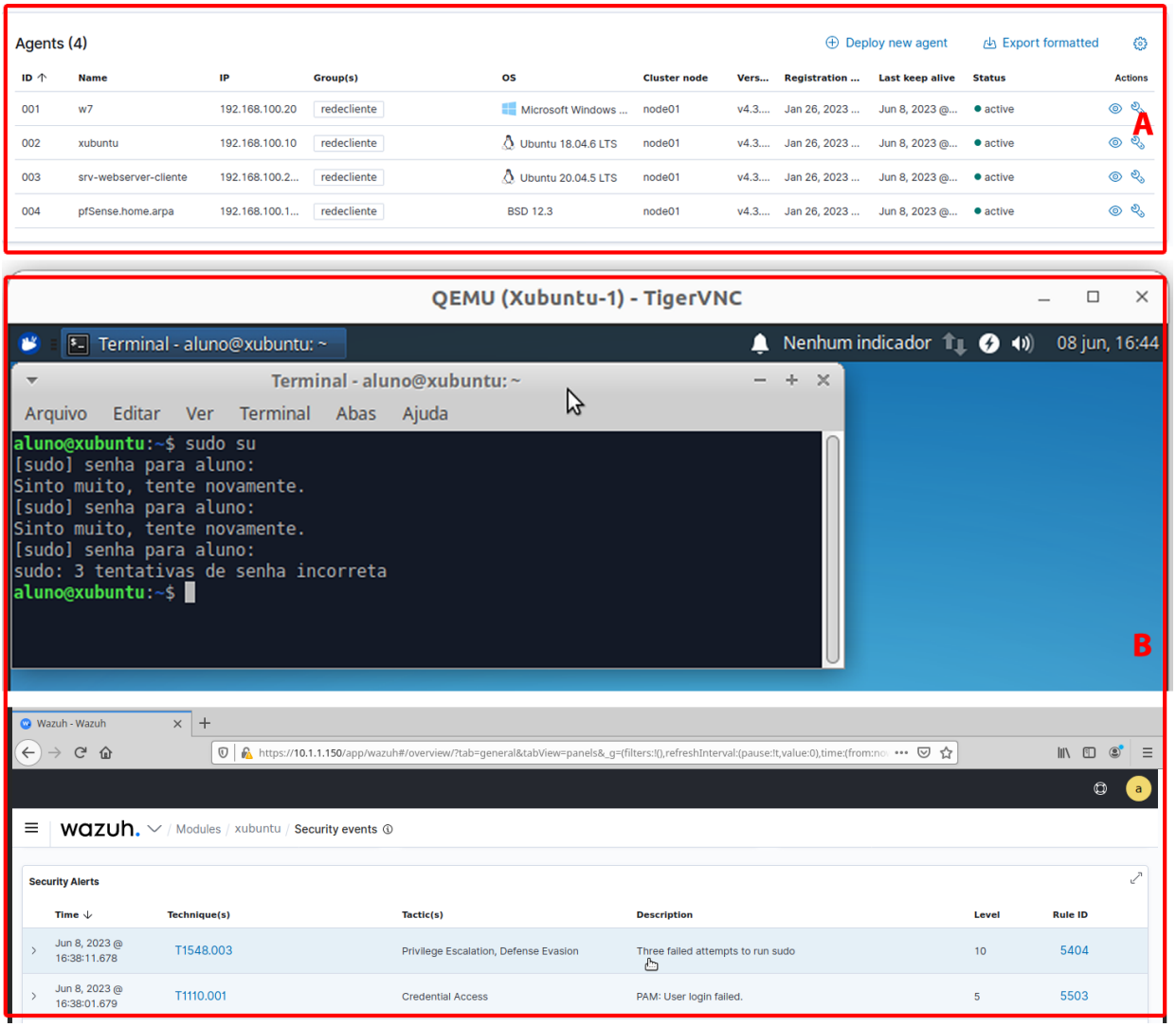


Figura 3.19: Wazuh server - agentes e eventos de segurança.

Também é possível gerar alertas por e-mail e outras modalidades de envio. Apesar de não possuir suporte específico para Telegram, o Wazuh permite integração com *Application Programming Interface* API externas, o que possibilita enviar alertas via API do Telegram com o uso de um *scrpit*. Em (SáNCHEZ, 2020) é possível encontrar um passo a passo de como realizar a configuração, neste caso o *script* foi adaptado para enviar mais informações e enviar apenas alertas com nível maior ou igual a 5, para evitar aletas excessivos. O *script* utilizado e as configurações estão nos apêndices deste trabalho da subseção 6.1.4.

A figura 3.20 mostra alertas gerados pelo Wazuh devido a erro em tentativas de efetuar login nas máquinas Xubuntu e Webserver.

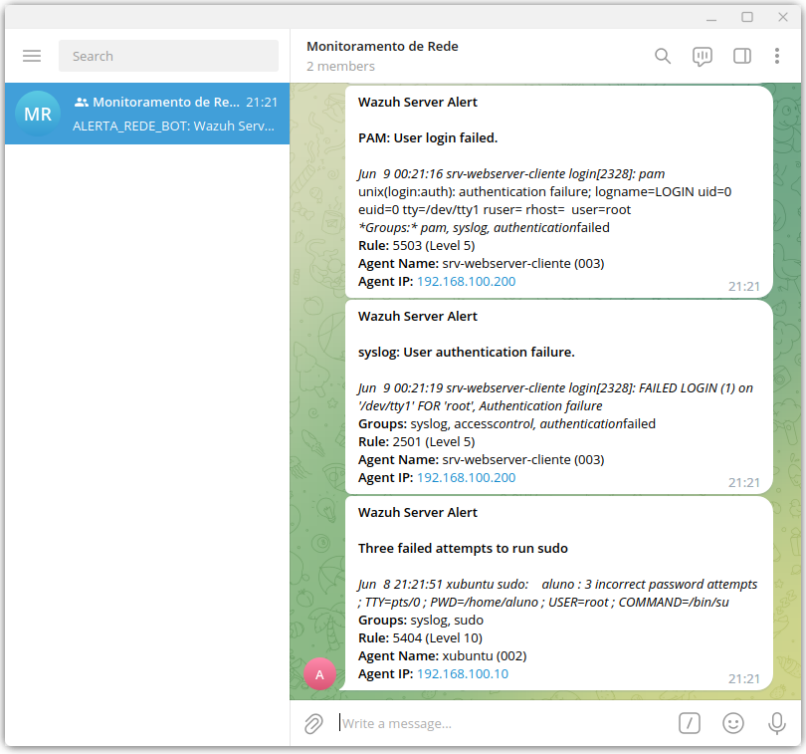


Figura 3.20: Alerta do Wazuh via Telegram.

4 EXPERIMENTO E RESULTADOS

Este capítulo apresenta a descrição dos testes realizados e os diferentes cenários utilizados, assim como aplicação dos testes em cada cenário e seus resultados.

4.1 TESTES DE REDE

Esta seção descreve as ferramentas utilizadas nos testes de conexão, estabilidade e taxas de trasmissão, discutindo os parâmetros e métricas utilizadas por cada ferramenta.

4.1.1 Teste de conexão

Utilitário PING - O PING é uma ferramenta para teste de conexão que usa mensagens do protocolo ICMP. Quando acionado ele envia para o endereço de destino, pacotes ICMP do tipo *Echo request*, quando o destino responde com ICMP *Echo reply*, significa que existe conexão entre origem e destino, pode acontecer de não chegar uma resposta ou chegar uma mensagem de erro, nestes casos existem códigos no ICMP que podem ajudar a depurar o problema (BASTA et al., 2014).

Ao final do envio e recebimento o PING exibe um resumo com estatísticas do teste, exibindo numero de pacotes enviados e recebidos, porcentagem de perda de pacotes, e o *Round Trip Time* RTT **"min"** mínimo, **"max"** máximo, **"avg"** média e **"mdev"** desvio padrão. O RTT é o tempo de ida e volta do pacote, ou seja, é o tempo de envio do pedido *Echo request* e o recebimento da resposta *Echo reply*. O parâmetro "mdev"é o desvio padrão do RTT, uma média da variação do RTT de cada envio. Quanto menor, melhor, um "mdev"muito alto pode significar uma conexão instável.

Os testes de ping utilizados neste trabalho foram realizados com a versão Linux do utilitário, usando o parâmetro -C, para especificar a quantidade de pacotes a serem enviados.

Exemplo:

```
ping 8.8.8.8 -c 100
```

4.1.2 Testes de velocidade

Ferramenta SIMET - Para testes de velocidade com servidores na *Internet* foi utilizado o SIMET, uma ferramenta do Núcleo de Informação e Coordenação do Ponto BR NIC.BR. Segundo (NIC-BR, 2023), a ferramenta mede a velocidade de *Download*, *Upload*, tempo do PING, JITTER e Perda de pacotes. Quando acionada a ferramenta escolhe um de seus servidores que esteja fora do provedor de quem está fazendo o teste, estabelece uma conexão e inicia as medições. Sobre os parâmetros utilizados pela ferramenta:

- **Velocidade de Download** - é a velocidade em que um arquivo pode ser baixado de um ponto na *Internet*;
- **Velocidade de Upload** - é a velocidade em que um arquivo pode ser enviado a um ponto na *Internet*;
- **PING ou Latência bidirecional** - é a medida de tempo para uma mensagem ir a um destino e voltar;
- **Jitter** - é a variação no atraso da transmissão de mensagens a um mesmo destino. É desejado que mensagens enviadas ao mesmo destino levem o mesmo tempo para chegar, um Jitter alto pode significar instabilidade na conexão;
- **Perda de pacotes** - é medida da quantidade de pacotes perdidos na transmissão.

A ferramenta pode ser acessada diretamente pelo navegador *Web* no endereço: <https://beta.simet.nic.br/>

Utilitário iPerf3 - Para os testes de taxa de transmissão entre as máquinas no cenário, foi utilizado o iPerf3, segundo o desenvolvedor da ferramenta (IPERF.FR, 2023), o iPerf3 é uma ferramenta de medição de banda e auxílio no ajuste de desempenho de rede. A ferramenta funciona no esquema cliente-servidor, onde é necessário acionar o iPerf nas duas pontas que se deseja realizar o teste, a ferramenta suporta vários parâmetros que podem ser usados para ajustes nos testes. Após o teste a ferramenta emite um relatório, as informações do relatório podem variar conforme o protocolo da camada de transporte utilizado no teste.

Para testes com ***Transmission Control Protocol*** **TCP**, são reportadas as seguintes métricas: *Interval, Transfer, Bitrate, Retransmitions, Cwnd*, para os dados enviados e para os recebidos.

Descrição das métricas da ferramenta iPerf3 segundo o desenvolvedor:

- **Interval** - Intervalo em segundos entre as medições em um teste;
- **Transfer** - Quantidade de dados transferidos em cada medição em MBytes;
- **Bitrate** - Taxa de transferência de dados por segundo em Mbits;

- **Retransmitions** - Quantidade de pacotes retransmitidos, acontece apenas com TCP por este fazer o controle de fluxo;
- **Cwnd** - Tamanho da janela de dados que podem ser enviados, faz parte do controle de congestionamento do TCP, é normalmente negociado entre cliente e servidor;

Neste trabalho o iPerf foi utilizado com os seguintes comandos e parâmetros:

```
#Para atuar como servidor
iperf3 -s

#Para se conectar ao servidor em TCP
iperf3 -n 50M -c  10.1.1.1
```

Parâmetros:

- **-s** - atuar como servidor;
- **-n** - numero de bytes transmitidos, usado para especificar quantos bytes devem ser enviados no teste;
- **-c** - se conectar a um servidor;

Ferramenta Rsync - É uma ferramenta de cópia de arquivos locais e remotos, suporta um grande número de parâmetros que controlam a execução da tarefa sobre o conjunto de arquivos a serem copiados. É largamente utilizado para realizar *backup* e espelhamentos, por ter a capacidade de fazer cópias incrementais e diferenciais (RSYNC-SAMBA.ORG, 2023).

A ferramenta permite acompanhar o processo de cópia e ao final emite estatísticas da tarefa. Neste trabalho o Rsync foi utilizado para realizar cópias de arquivos com tamanhos pré-definidos para testar a taxa de transferência e qualidade da conexão. As métricas informadas ao final da transferência são: *bytes* enviados, *bytes* recebidos e taxa de transferência em *bytes*/segundo. Segue o comando utilizado, os parâmetros e seus significados.

```
#Copiar arquivo da maquina local para uma maquina remota
rsync -avhP arquivo_100M usuario@192.168.1.17:/home/usuario
```

Parâmetros:

- **-a** - utilizada para recursividade dentro de uma pasta e para preservar algumas propriedades do arquivo, como permissões, proprietário, grupo e outras;

- **-v** - *verbose*, utilizada para obter informações sobre a tarefa;
- **-h** - mostra números em formato de fácil leitura para humanos;
- **-P** - mostra progresso durante a transferência.

Ferramenta nmap - é uma ferramenta de código aberto utilizado para varredura e descoberta de rede, pode também ser utilizado para inventário de rede. A aplicação envia pacotes pela rede para descobrir quais máquinas estão ativas e quais serviços estão hospedados nessas máquinas. A ferramenta possui grande quantidade parâmetros que possibilitam diversos tipos de testes (INSECURE.ORG, 2023).

4.2 CENÁRIOS E TESTES

Esta seção descreve os teses realizados em cada cenário e seus resultados.

4.2.1 Testes de rede na máquina hospedeira

Seguem testes de taxa de transmissão e conexão com a *Internet* na máquina hospedeira pra que sirvam de parâmetro comparativo, nos testes com a rede virtualizada. A figura 4.1 mostra resultados dos testes realizados na máquina hospedeira.

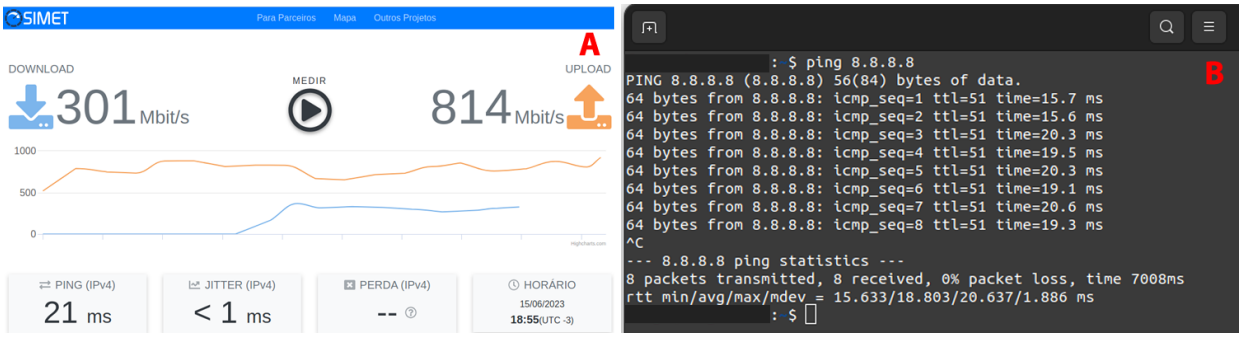


Figura 4.1: Cenário 1 Topologia - Testes de rede máquina hospedeira.

Resultados dos testes:

- Figura 4.1 - detalhe A - mostra a saída do teste de taxa de transmissão: *Download* 301Mbits, *Upload* 814Mbits, média de PING 20ms, JITTER 1ms.
- Figura 4.1 - detalhe B - mostra a saída do teste de conexão com PING, RTT médio de 18.8ms e mdev de 1.8ms.

4.2.2 Cenário 1

Cenário 1 - Duas redes virtualizadas na mesma máquina - Cenário simples, com duas redes virtualizadas na mesma máquina.

A máquina utilizada para hospedar a virtualização foi a M1 da tabela 3.1. A tabela 4.1 mostra as máquinas presentes no cenário e a figura 4.2 mostra a topologia do cenário.

Elemento	Rede
Zabbix	Rede de Serviços
Linux	Rede de Serviços
pfsense	Rede de Serviços
DD-WRT	Rede Cliente
pfsense	Rede Cliente
Linux	Rede Cliente

Tabela 4.1: Elementos do Cenário 1.

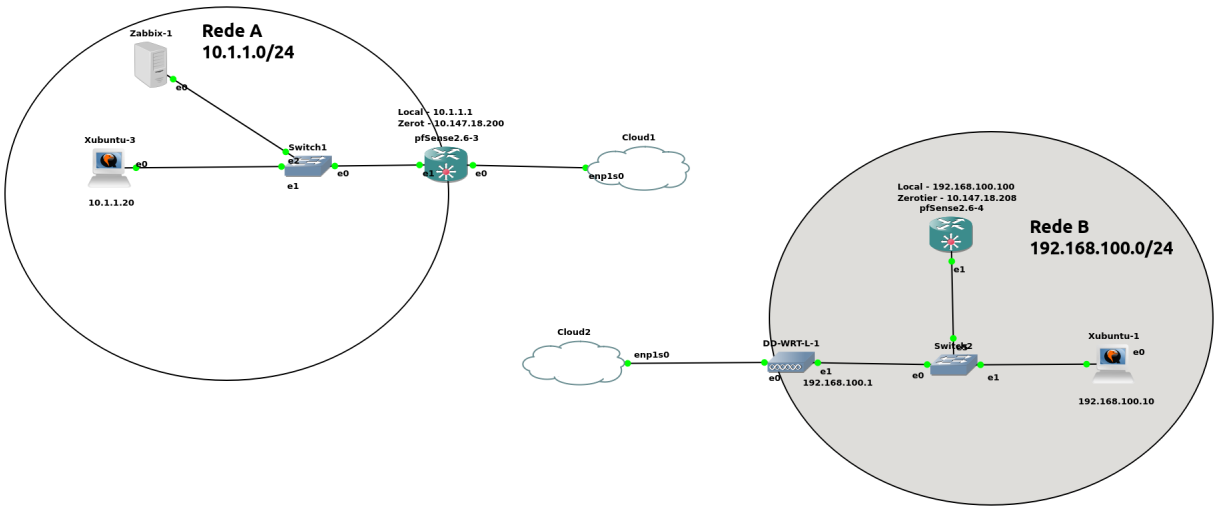


Figura 4.2: Cenário 1 Topologia - 2 redes virtualizadas na mesma máquina.

Teste transmissão

Teste de taxa de transmissão pela *Internet* com a ferramenta SIMET. A figura 4.3 mostra a saída do teste realizado na máquina Linux da rede de serviços e na máquina Linux da rede cliente.

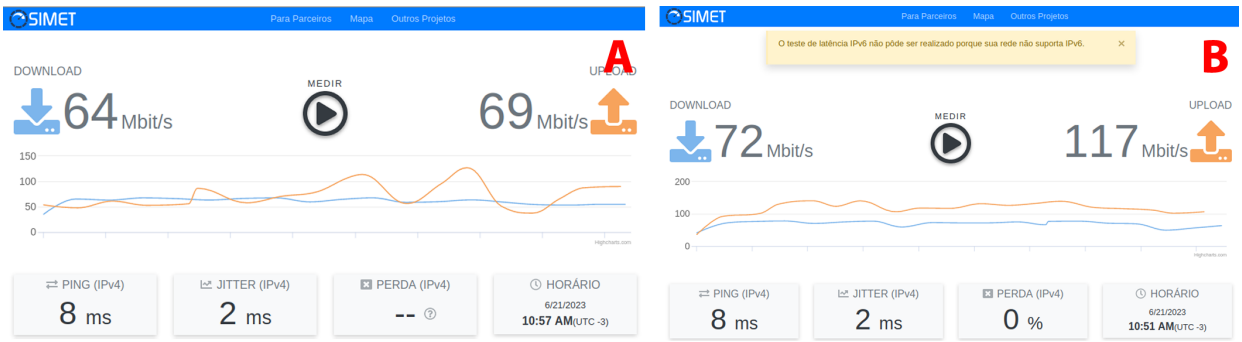


Figura 4.3: Cenário 1 teste de velocidade de acesso à *Internet*.

Detalhes do teste com SIMET:

- figura 4.3 - detalhe A - mostra a saída do teste realizado na máquina Linux da rede de serviços;
- figura 4.3 - detalhe B - mostra a saída do teste realizado na máquina Linux da rede de cliente;

O maior de ponto a se considerar sobre esse resultado, é que claramente se vê que o cenário virtualizado não tem o mesmo desempenho de rede da máquina hospedeira, o que já era esperado, visto que o acesso à *Internet* é feito passando por um *gateway* também virtualizado e com recursos de *hardware* e de rede compartilhados.

Teste de PING

Caminho dos pacotes entre as redes

O caminho dos pacotes da rede de serviços para a rede cliente passa pela rede virtual ZeroTier, que funciona como túnel para interligar as duas redes. O caminho traçado dos pacotes pode ser visto na figura 4.4.

```
Terminal - aluno@xubuntu: ~
Arquivo  Editar  Ver  Terminal  Abas  Ajuda
aluno@xubuntu:~$ ip -o -4 a  | grep inet | cut -d" " -f1-9
1: lo    inet 127.0.0.1/8 scope host
2: ens4    inet 10.1.1.20/24 brd 10.1.1.255
aluno@xubuntu:~$                                                    A
aluno@xubuntu:~$ ip route
default via 10.1.1.1 dev ens4 proto static
10.1.1.0/24 dev ens4 proto kernel scope link src 10.1.1.20          B
aluno@xubuntu:~$
aluno@xubuntu:~$ traceroute 192.168.100.10
traceroute to 192.168.100.10 (192.168.100.10), 30 hops max, 60 byte packets
 1  _gateway (10.1.1.1)  0.476 ms  0.519 ms  0.491 ms
 2  10.147.18.208 (10.147.18.208)  2.139 ms  2.822 ms  3.084 ms      C
 3  192.168.100.10 (192.168.100.10)  3.161 ms  3.256 ms  3.181 ms
aluno@xubuntu:~$
```

Figura 4.4: Cenário 1 Teste de traceroute da rede de serviços para a rede cliente.

Detalhes da figura 4.4:

- Figura 4.4 - detalhe A - mostra as interfaces de rede da máquina Linux da rede de serviços, sendo a interface **ens4** a interface de saída com endereço IP 10.1.1.20/24;
- Figura 4.4 - detalhe B - mostra a rota padrão da rede apontando para a máquina 10.1.1.1 que é o pfsense da rede de serviços e funciona como *gateway* para a *Internet* e para a rede ZeroTier;
- Figura 4.4 - detalhe C - mostra o caminho percorrido pelos pacotes da rede de serviços até a máquina Linux da rede cliente, passando pelo *gateway* na rede 10.1.1.0/24, em seguida pelo *gateway* ZeroTier na rede cliente e chegando a máquina Linux da rede cliente.

Teste de conexão

Na máquina Linux da rede de serviços, endereço IP 10.1.1.20 foram efetuados 3 disparos de ping com 100 pacotes cada, para a máquina Linux na rede cliente com endereço IP 192.168.100.10. Comando:

```
#Ping da maquina Linux servicos para a maquina Linux cliente
ping 192.168.100.10 -c 100
```

A figura 4.5 mostra a saída dos três disparos de ping realizados da rede de serviços para a rede cliente.

```
--- 192.168.100.10 ping statistics ---
100 packets transmitted, 100 received, 0% packet loss, time 99160ms
rtt min/avg/max/mdev = 1.886/2.572/5.250/0.555 ms
```

```
--- 192.168.100.10 ping statistics ---
100 packets transmitted, 100 received, 0% packet loss, time 99158ms
rtt min/avg/max/mdev = 1.984/2.583/8.284/0.773 ms
```

```
--- 192.168.100.10 ping statistics ---
100 packets transmitted, 100 received, 0% packet loss, time 99165ms
rtt min/avg/max/mdev = 1.684/2.589/5.109/0.537 ms
```

Figura 4.5: Cenário 1 Teste de ping da rede de serviços para a rede cliente.

A tabela 4.2 mostra as médias das métricas dos três disparos:

Perda	RTT min	RTT avg	RTT max	RTT mdev
0,000%	1,8460ms	2,5793ms	6,21433ms	0,6217ms

Tabela 4.2: Cenário 1 resultados PING.

Considerando que para a rede de serviços alcançar máquinas na rede cliente, é necessário passar pelo *gateway* ZeroTier. Os pacotes ICMP Echo Request do ping saíram da rede de serviços passaram pela rede ZeroTier na *Internet*, e chegaram à rede de cliente, os pacotes ICMP Echo Reply fizeram o caminho inverso, e ainda assim, os resultados obtidos no teste ficam abaixo dos valores de ping da máquina hospedeira para a *Internet*, com destaque para o mdev que ficou a baixo de 0 ms. Assim sendo o teste indica uma boa conectividade entre os dispositivos nas duas redes.

Teste com iPerf

No teste com iPerf a máquina de Linux da rede cliente funcionou como servidor e a máquina Linux da rede de serviços como cliente, foram efetuados três testes de transferência, com arquivos de 50Mbytes em TCP. Seguem os comandos:

```
#Maquina Linux rede cliente, como servidor
iperf3 -s

#Teste em TCP - Maquina Linux rede servicos, como cliente
iperf3 -n 50M -c 192.168.100.10
```

A figura 4.6 mostra os resultados dos testes.

```
[ ID] Interval           Transfer     Bandwidth       Retr
[  4]   0.00-22.57  sec  50.0 MBytes  18.6 Mbits/sec   18             sender
[  4]   0.00-22.57  sec  49.4 MBytes  18.3 Mbits/sec                  receiver
```

```
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  4]   0.00-22.14  sec  50.1 MBytes  19.0 Mbits/sec   14             sender
[  4]   0.00-22.14  sec  49.4 MBytes  18.7 Mbits/sec                  receiver
```

```
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  4]   0.00-22.13  sec  50.2 MBytes  19.0 Mbits/sec   23             sender
[  4]   0.00-22.13  sec  49.5 MBytes  18.8 Mbits/sec                  receiver
```

Figura 4.6: Cenário 1 Teste de transferência de arquivo com iPerf.

A tabela 4.3 mostra os resultados obtidos:

Bandwidth Mbits/sec		**Retransmitions**
Sender	Receiver	Packets
18,8667	18,6000	18,3333

Tabela 4.3: Cenário 1 médias das métricas iPerf.

Controle fluxo TCP

O protocolo TCP atua na camada de transporte e é responsável, entre outras coisas, pelo estabelecimento da conexão entre servidor e cliente. O TCP realiza controle do fluxo de dados enviados e recebidos, o protocolo de Janelas Deslizantes faz parte desse controle de fluxo, resumidamente o que ele faz é negociar a quantidade de dados que o servidor pode enviar para o cliente. Para tentar maximizar o uso de banda da conexão, o servidor tenta aumentar gradualmente a quantidade de dados enviados, quando percebe que alguns pacotes não estão chegando no destino, ele reduz a quantidade de dados enviados e recomeça o aumento gradual novamente (COMER, 2016). Esse comportamento pode ser observado na saída do iPerf3 na figura 4.7, onde é possível visualizar a variação no tamanho da janela na coluna **Cwnd**.

A figura 4.6 mostra os valores da janela de envios.

```
Connecting to host 192.168.100.10, port 5201
[  4] local 10.1.1.3 port 55932 connected to 192.168.100.10 port 5201
[ ID] Interval           Transfer     Bandwidth       Retr  Cwnd
[  4]   0.00-1.00   sec  2.86 MBytes  24.0 Mbits/sec    4   94.7 KBytes
[  4]   1.00-2.00   sec  2.30 MBytes  19.3 Mbits/sec    0    117 KBytes
[  4]   2.00-3.00   sec  2.24 MBytes  18.8 Mbits/sec    1   94.7 KBytes
[  4]   3.00-4.00   sec  2.17 MBytes  18.2 Mbits/sec    0    112 KBytes
[  4]   4.00-5.00   sec  2.24 MBytes  18.8 Mbits/sec    0    126 KBytes
[  4]   5.00-6.00   sec  2.24 MBytes  18.8 Mbits/sec    2    109 KBytes
[  4]   6.00-7.00   sec  2.17 MBytes  18.3 Mbits/sec    0    122 KBytes
[  4]   7.00-8.00   sec  2.24 MBytes  18.8 Mbits/sec    1   91.9 KBytes
[  4]   8.00-9.00   sec  2.17 MBytes  18.2 Mbits/sec    0    107 KBytes
[  4]   9.00-10.00  sec  2.24 MBytes  18.8 Mbits/sec    0    123 KBytes
```

Figura 4.7: Cenário 1 janela de envio TCP saída iPerf.

Os resultados obtidos nos três testes, mostram valores próximos, variando nas casas dos decimais, o que pode significar estabilidade na conexão entre as redes, mesmo tendo que passar pelo túnel. Apesar de alguns pacotes retransmitidos.

Teste com Rsync

No teste com rsync a máquina de Linux da rede cliente funcionou como origem e a máquina Linux da rede de serviços como destino, foram efetuados três testes de transferência, com arquivos de 100Mbytes. Seguem os comandos:

```
1    #Maquina Linux rede cliente como destino dos dados
     rync -avhP arquivo_100M aluno@192.168.100.10:/home/aluno
```

A figura 4.8 mostra os resultados dos testes.

```
aluno@LinuxServicos:~$ rsync -avhP arquivo_100M  aluno@192.168.100.10:/home/aluno
aluno@192.168.100.10's password:
sending incremental file list
arquivo_100M
        104.86M 100%    2.17MB/s    0:00:46 (xfr#1, to-chk=0/1)

sent 104.88M bytes  received 35 bytes  2.08M bytes/sec
total size is 104.86M  speedup is 1.00
aluno@LinuxServicos:~$
aluno@LinuxServicos:~$ rsync -avhP arquivo_100M  aluno@192.168.100.10:/home/aluno
aluno@192.168.100.10's password:
sending incremental file list
arquivo_100M
        104.86M 100%    2.22MB/s    0:00:45 (xfr#1, to-chk=0/1)

sent 104.88M bytes  received 35 bytes  2.08M bytes/sec
total size is 104.86M  speedup is 1.00
aluno@LinuxServicos:~$ rsync -avhP arquivo_100M  aluno@192.168.100.10:/home/aluno
aluno@192.168.100.10's password:
sending incremental file list
arquivo_100M
        104.86M 100%    2.21MB/s    0:00:45 (xfr#1, to-chk=0/1)

sent 104.88M bytes  received 35 bytes  1.89M bytes/sec
total size is 104.86M  speedup is 1.00
aluno@LinuxServicos:~$
```

Figura 4.8: Cenário 1 saída transferência com rsync.

A tabela 4.4 mostra as médias dos resultados obtidos:

Taxa de transmissão		
Mínimo	Média	Máximo
1.89 MBytes/sec	2.49 MBytes/sec	2.8 MBytes/sec

Tabela 4.4: Cenário 1 resultados rsync.

As taxas de transferências obtidas nos três testes com rsync, mostram valores próximos, apesar de não apresentar uma taxa de transmissão alta, ela aparenta ter estabilidade.

Conexão com Zabbix

A figura 4.9 mostra o painel de *hosts* monitorados pelo Zabbix, como pode ser observado na coluna *Availability* o ícone que representa o agente Zabbix na máquina cliente está na cor verde, representando que a máquina está acessível para o Zabbix, neste caso, estão sendo monitoradas as máquinas Xubuntu e pfSense na rede cliente. O que significa que a conexão das duas redes via túnel ZeroTier em um esquema *site-to-site* é suficiente para permitir que o servidor monitore os dispositivos na rede cliente remota.

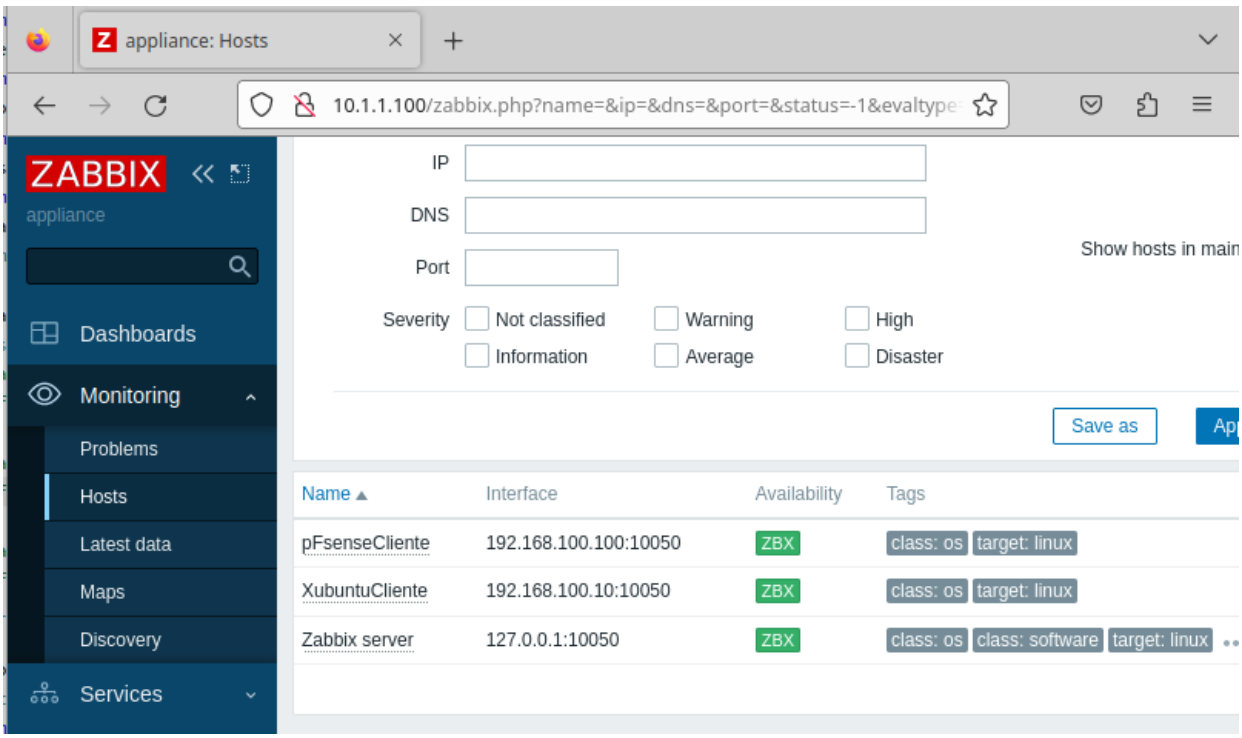


Figura 4.9: Cenário 1 tela de monitoramento Zabbix.

Em suma o cenário 1, virtualizando as duas redes na mesma máquina, apesar de simples, funcionou bem e apresentou uma conexão estável, entre as duas redes, passando por um túnel na *internet* e ainda assim, possibilitando acesso remoto, transferência de arquivos e monitoramento das estações remotas.

4.2.3 Cenário 2

Cenário 2 - Cenário misto, uma rede física e uma rede virtualizada - neste cenário, a rede cliente é comporta por máquinas físicas, incluindo o roteador, enquanto a rede de serviços é virtualizada. A tabela 4.5 apresenta os elementos em cada rede do cenário e sua respectiva máquina hospedeira, a especificação correspondente de cada equipamento pode ser visto na tabela 3.1.

Elemento	Rede	Máquina hospedeira
Zabbix	Rede de Serviços	M1
Linux	Rede de Serviços	M1
pfsense	Rede de Serviços	M1
DD-WRT	Rede Cliente	R1
pfsense	Rede Cliente	M5
Linux	Rede Cliente	M2
Windows	Rede Cliente	M3

Tabela 4.5: Elementos do Cenário 2.

A figura 4.10, mostra a topologia utilizada no cenário 2, com identificação dos dispositivos utilizados.

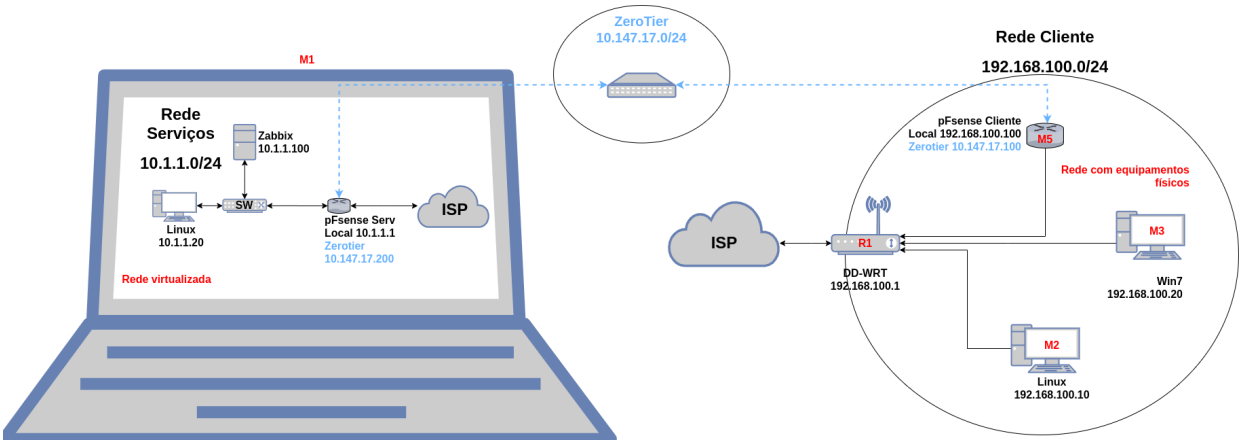


Figura 4.10: Cenário 2 Topologia - 1 rede virtualizada e 1 rede física.

A figura 4.11 mostra uma foto da máquina M1 hospedeira da rede de serviços virtualizada.

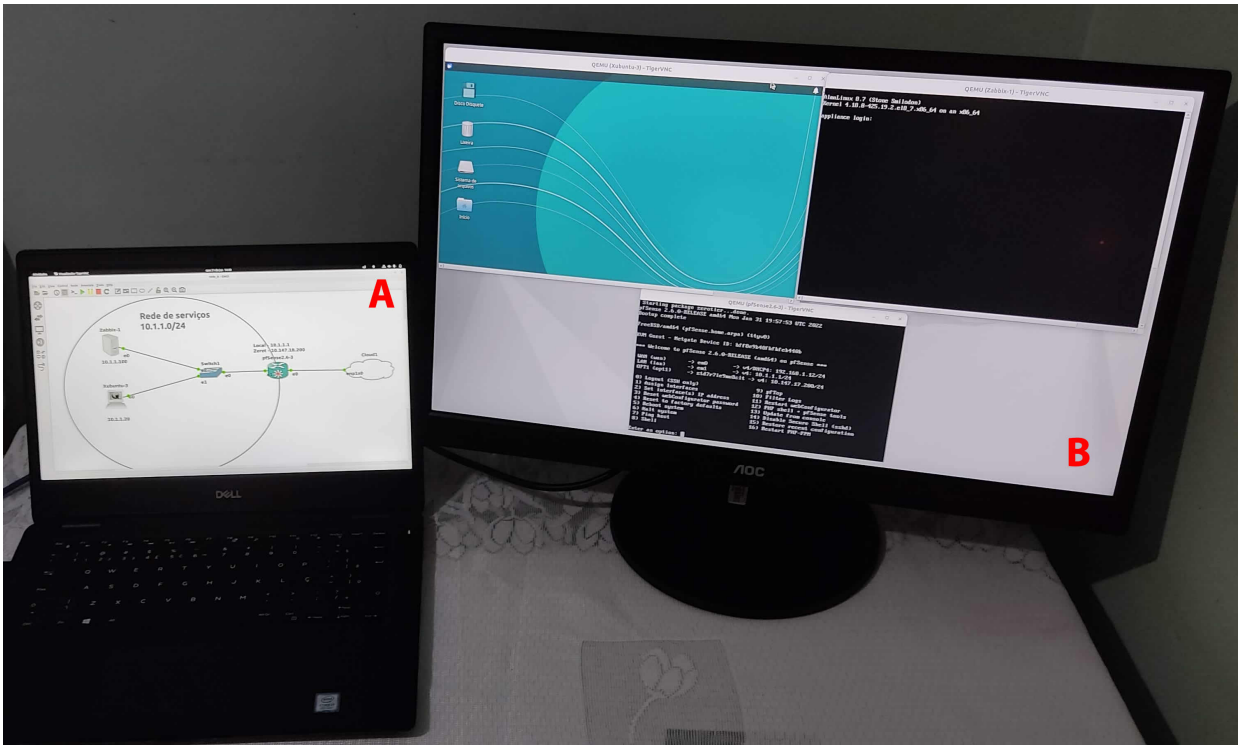


Figura 4.11: Cenário 2 - Foto rede virtualizada, máquina M1 com dois monitores

- figura 4.11 - detalhe A - mostra a máquina M1 da tabela 3.1, no monitor da máquina é possível ver o cenário do GNS3 com os equipamentos virtuais.
- figura 4.11 - detalhe B - mostra um monitor extra conectado à máquina M1, onde é possível ver a tela das máquinas virtuais da rede.

A figura 4.12 mostra uma foto das máquinas que compõe a rede cliente física.

Figura 4.12: Cenário 2 - Foto rede com máquinas físicas

- figura 4.12 - detalhe A - mostra a máquina M3 onde foi instalado o sistema operacional Windows.
- figura 4.12 - detalhe B - mostra o equipamento R1 onde foi instalado o *firmware* DD-WRT.
- figura 4.12 - detalhe C - mostra a máquina M5 onde foi instalado o sistema operacional pFsense da rede cliente, responsável pela conexão com o túnel ZeroTier.
- figura 4.12 - detalhe D - mostra a máquina M2 onde foi instalado um sistema operacional Linux.

A descrição dos equipamentos físicos pode ser entrada na tabela 3.1.

Teste de transmissão

Teste de taxa de transmissão pela Internet com a ferramenta SIMET. A figura 4.3 mostra a saída do teste realizado na máquina Linux da rede de serviços virtualizada, e na máquina Linux da rede cliente, máquina física.

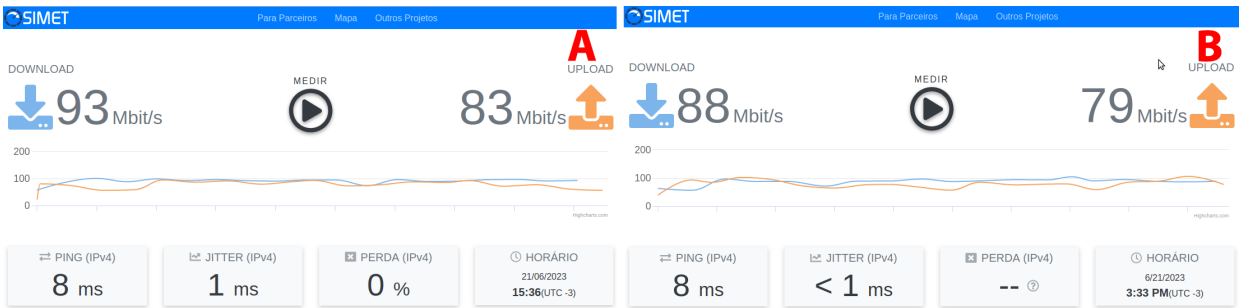


Figura 4.13: Cenário 2 - Foto rede com máquinas físicas

- figura 4.13 - detalhe A - mostra a saída do teste realizado na máquina Linux da rede de serviços, máquina virtualizada.
- figura 4.13 - detalhe b - mostra a saída do teste realizado na máquina Linux da rede de cliente, máquina física.

Dois pontos devem ser considerados neste teste, primeiro, na rede de serviços as taxas de transmissões continuam abaixo das taxas da rede hospedeira, que podem ser vistas na figura 4.3, isso se deve em parte pela virtualização dos dispositivos, e pela quantidade de recursos alocados para cada equipamento. O segundo ponto é sobre as taxas obtidas no teste com a máquina física na rede cliente, neste caso, o roteador de saída da rede R1 é um dispositivo com portas 10/100, ou seja, portas com taxas de transmissão de no máximo 100 Mbits/segundo. Assim sendo, as taxas obtidas de 93Mbits para *download* e 79Mbist para *upload*, podem ser consideradas boas, visto que estão próximas do limite do roteador de borda.

Teste de PING

Caminho dos pacotes entre as redes

O caminho dos pacotes da rede de serviços virtualizada, para a rede cliente com máquinas físicas, passa pelo túnel ZeroTier que interliga as redes. O caminho traçado dos pacotes pode ser visto na figura 4.14.

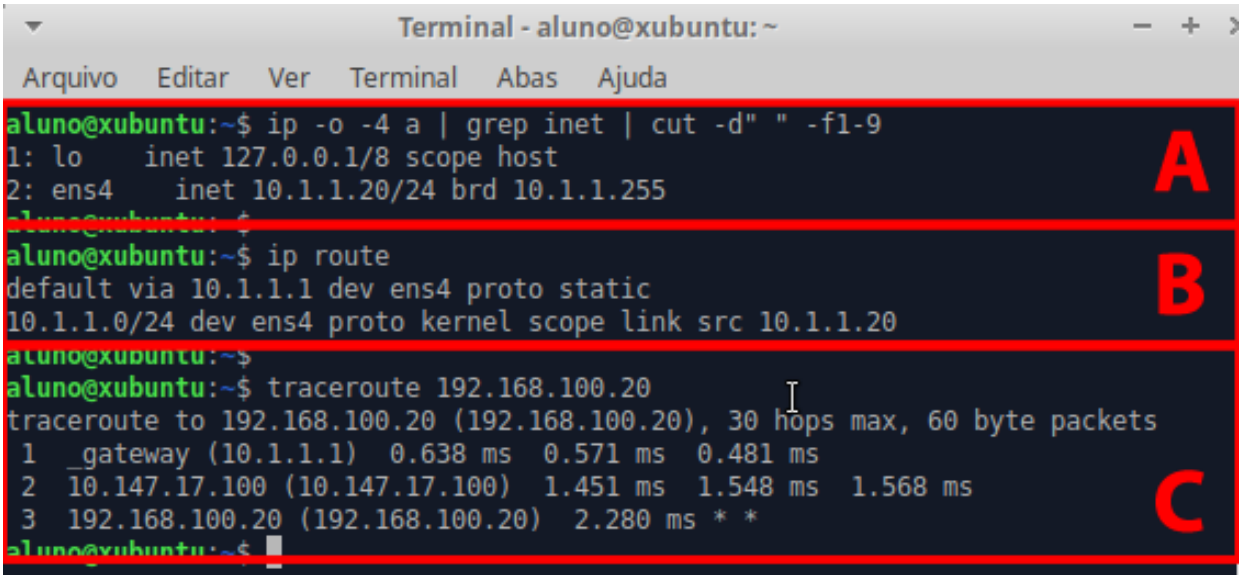


Figura 4.14: Cenário 2 Teste de traceroute da rede de serviços para a rede cliente.

Detalhes da figura 4.14:

- figura 4.14 - detalhe A - mostra as interfaces de rede da máquina Linux da rede de serviços, sendo a interface **ens4** a interface de saída com endereço IP 10.1.1.20/24;

- figura 4.14 - detalhe B - mostra a rota padrão da rede apontando para a máquina 10.1.1.1 que é o pfsense da rede de serviços e funciona como *gateway* para a *Internet* e para a rede ZeroTier;
- figura 4.14 - detalhe C - mostra o caminho percorrido pelos pacotes da rede de serviços até a máquina Linux da rede cliente, passando pelo *gateway* na rede 10.1.1.0/24, em seguida pelo *gateway* ZeroTier na rede cliente e chegando a máquina Windows da rede cliente.

Teste de conexão

Na máquina Linux da rede de serviços, endereço IP 10.1.1.10 foram efetuados disparos de ping com 100 pacotes cada, para a máquina Linux na rede cliente com endereço IP 192.168.100.10, os disparos foram enviados a cada 15 minutos no período de 14h45min até 22h15min do mesmo dia. Foi criado um pequeno *script* para fazer os disparos e salvar a saída em um arquivo de log, um agendamento no cron foi criado para executar o *scritp*. O *script* pode ser visto nos apêndices deste trabalho subseção 6.2.1.

Comando:

```
#Agendamento no cron para realizar os disparos de ping
nano /etc/crontrab
#Adicionar a linha do agen
*/15 * * * * aluno /home/aluno/testes.sh
```

Ping perda de pacotes - foram realizados 31 disparos de 100 pacotes cada, ainda assim, não foi registrada a perda de pacotes. O gráfico apresentado na figura 4.15 gerado com o log dos disparos, mostra que a perda se mantêm em zero em todos os disparos.

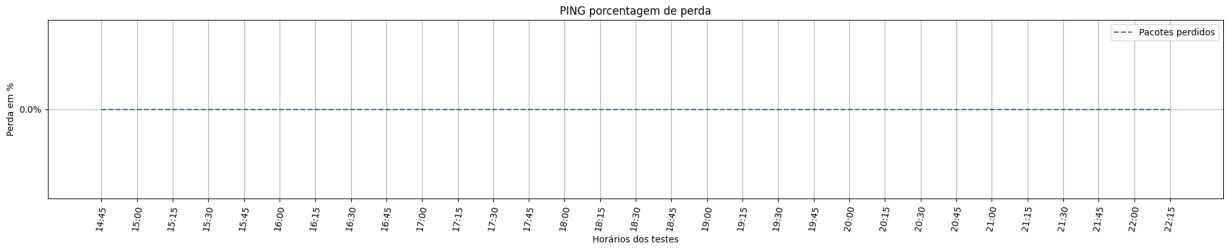


Figura 4.15: Cenário 2 Teste de PING - Perda de pacotes.

Ping RTT - foram realizados 31 disparos de 100 pacotes cada, nestes testes, o RTT médio variou de 2ms à 2.8ms, no ping entre a rede cliente e rede de serviços, considerando que o RTT médio da rede hospedeira para o servidor do Google na *internet* foi de 18.6ms como pode ser visto na figura 4.1, os valores de RTT obtidos nos testes podem ser considerados baixos e aceitáveis. O gráfico presentador na figura 4.16 gerado com o log dos disparos, mostra a variação do RTT nos testes.

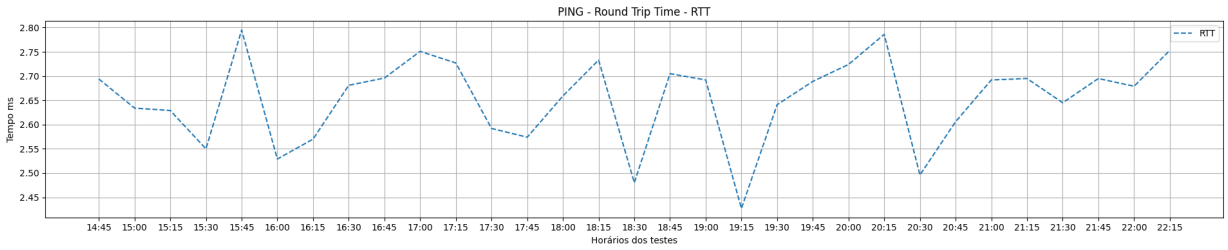


Figura 4.16: Cenário 2 Teste de PING - RTT.

Teste com iPerf

Nos testes com iPerf3, a máquina Linux na rede cliente funcionou como servidor e a máquina Linux na rede de serviços funcionou como cliente, enviado os dados. Foram feitos 31 testes de transmissão com configurações padrão do iPerf3 e sem especificação de tamanho de arquivo. O gráfico presente na figura 4.17 mostra a variação das taxas de transmissão obtidas nos testes.

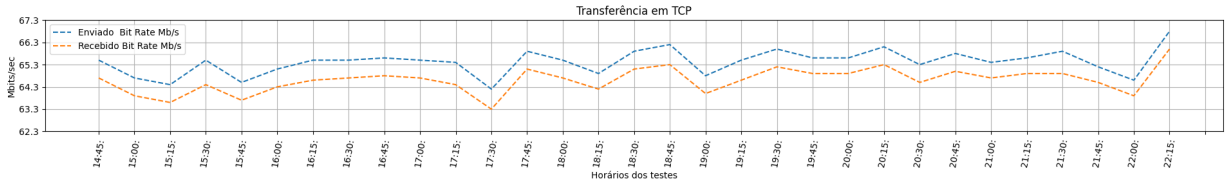


Figura 4.17: Cenário 2 Teste iPerf3 - Taxa de transmissão.

O gráfico na figura 4.18, mostra as taxas de retransmissão de pacotes obtidas durante os testes.

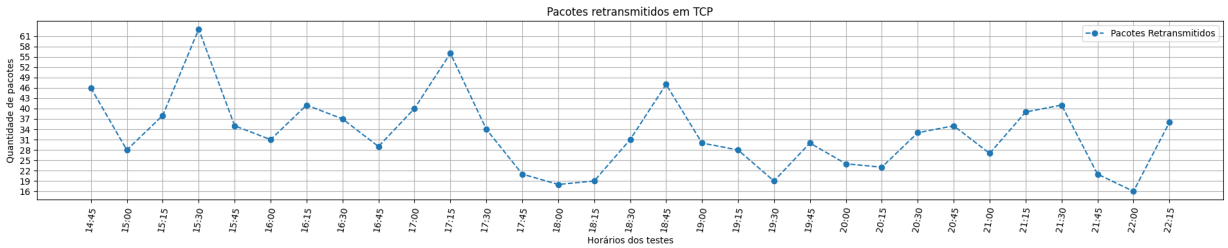


Figura 4.18: Cenário 2 Teste iPerf3 - Pacotes retransmitidos.

A tabela 4.6 mostra dos resultados obtidos:

Bandwidth Mbits/sec		**Retransmitions**
Sender	Receiver	Packets
65,4193	64,6064	32

Tabela 4.6: Cenário 2 médias das taxas iPerf3.

Considerando a limitação inserida na rede cliente pelo roteador de borda, com taxa máxima de transmissão nominal de 100Mbits/seg, as taxas de transmissão obtidas nos testes entre as

redes, estão em níveis aceitáveis para os objetivos de acesso remoto e monitoramento, enquanto a quantidade de pacotes retransmitidos se manteve baixa.

Teste com Rsync

No teste com rsync a máquina de Linux da rede de serviços funcionou como origem e a máquina Linux da rede cliente como destino, foram efetuados três testes de transferência, com arquivos de 100Mbytes. Seguem os comandos:

```
#Maquina Linux rede de cliente como destino dos dados
rync -avhP arquivo_100M aluno@192.168.100.10:/home/aluno
```

A figura 4.19 mostra os resultados dos testes.

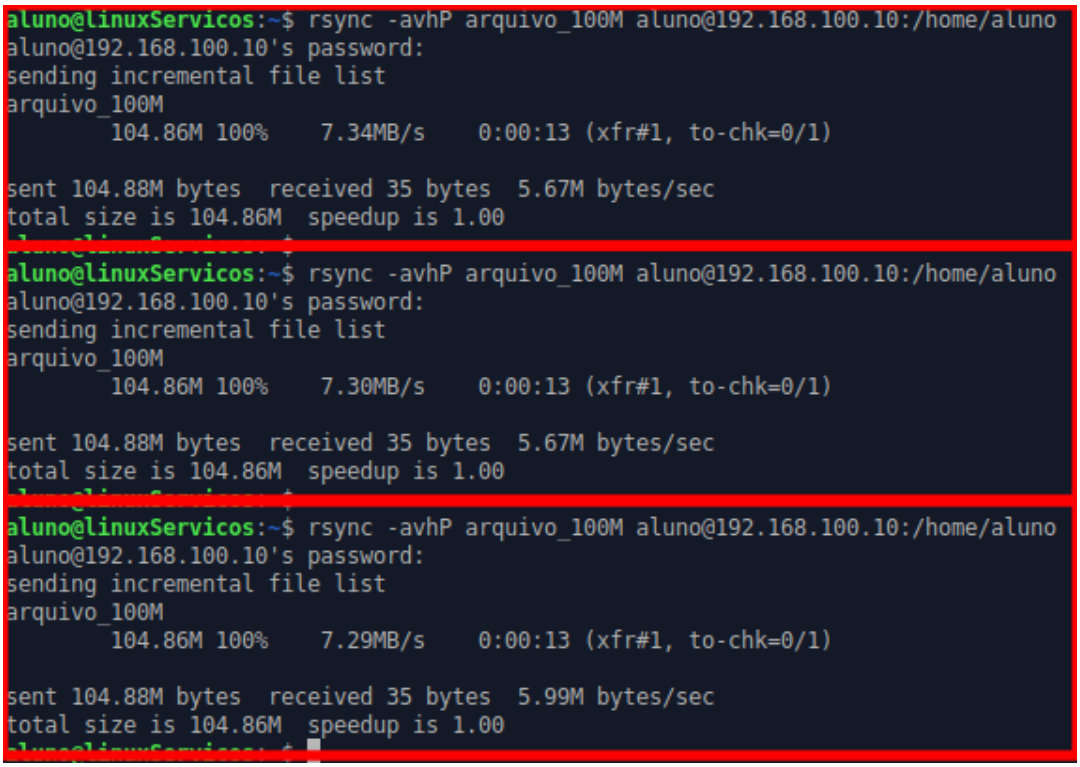


Figura 4.19: Cenário 2 saída transferência com rsync.

A tabela 4.7 mostra as médias dos resultados obtidos:

Taxa de transmissão		
Mínimo	Média	Máximo
5.67 MBytes/sec	5.77 MBytes/sec	5.99 MBytes/sec

Tabela 4.7: Cenário 2 resultados rsync.

As taxas de transferências obtidas nos três testes com rsync, mostram valores próximos, o que

pode significa estabilidade, as taxas estão dentro do esperado para uma rede com largura de banda nominal de 100Mbits, sendo o caso da rede cliente.

Conexão com Zabbix

A figura 4.20 mostra o painel de *hosts* monitorados pelo Zabbix, como pode ser observado na coluna *Availability* as máquinas monitoras estão *on-line*, neste cenário, são monitoradas as máquinas Linux, Windows e pfSense na rede cliente.

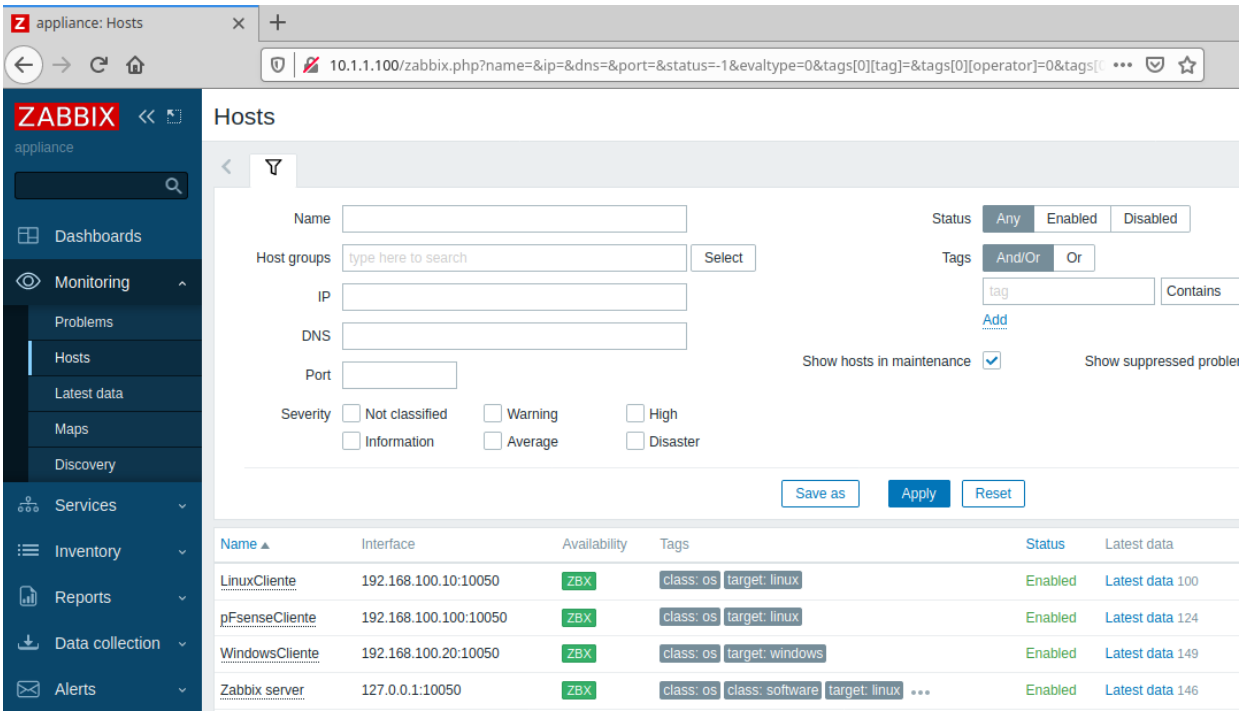


Figura 4.20: Cenário 2 tela de monitoramento Zabbix.

Acesso remoto

O acesso remoto via linha de comando nas máquinas Linux nas redes remotas é testado no durante os testes de transferência de dados com rsync, pois esta aplicação utiliza ssh para fazer transferências.

Para testar o acesso remoto à máquina Windows na rede cliente, foi utilizado na máquina Linux da rede de serviços, com o aplicativo **Remmina** que permite o uso de diversos protocolos de acesso remoto, dentre eles o RDP. A figura 4.21 mostra a tela da aplicação Remmina durante o acesso remoto à máquina Windows.

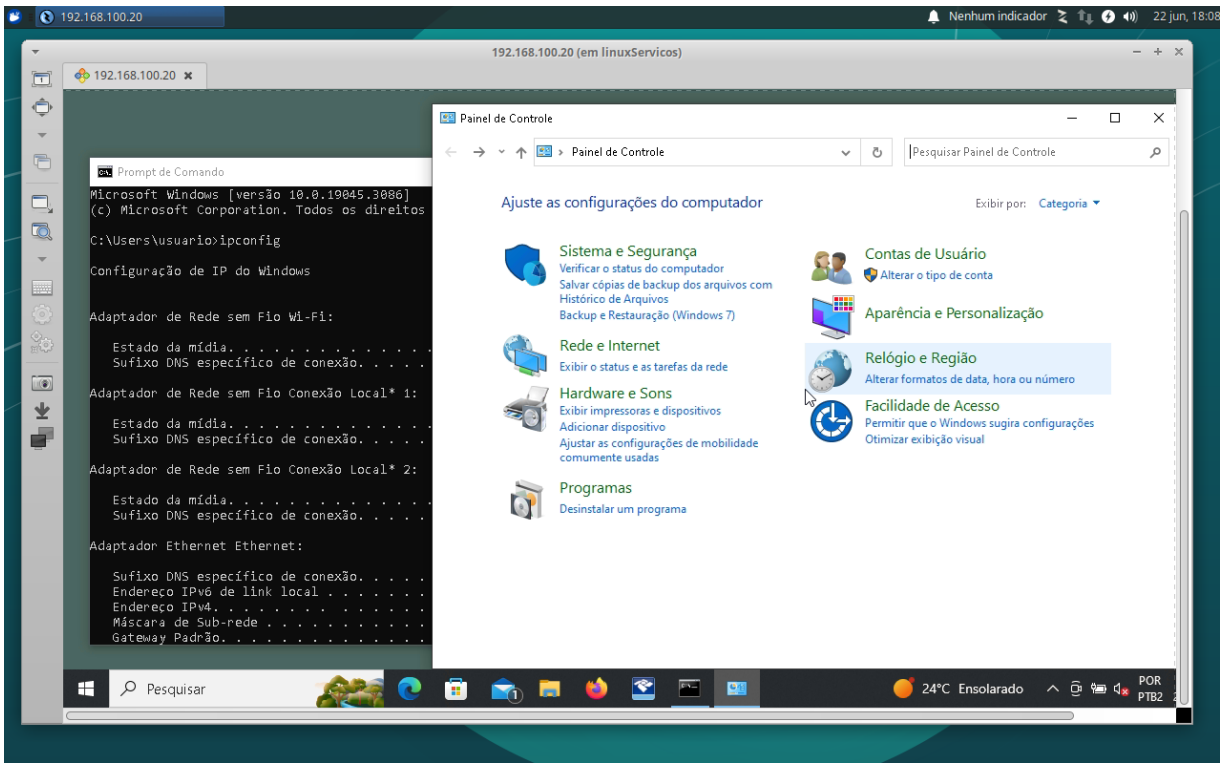


Figura 4.21: Cenário 2 acesso remoto Windows.

Os testes realizados no cenário 2, misto de virtual e físico, apresentaram estabilidade na rede e taxas de transferências melhores que as obtidas no cenário 1, possivelmente pela maior quantidade de *hardware* independente e menor compartilhamento de recursos. Os resultados mostram ser possível tanto acessar como monitorar a rede remotamente via túnel ZeroTier.

4.2.4 Cenário 3

Cenário 3 - totalmente físico - tanto a rede de serviços quanto a rede cliente é comporta por máquinas físicas, as redes estão geograficamente separadas, estando cada uma em uma residência, em quadras diferentes da mesa cidade. A tabela 4.8 apresenta os elementos em cada rede do cenário e sua respectiva máquina hospedeira, a especificação correspondente de cada equipamento hospedeiro pode ser visto na tabela 3.1.

A figura 4.22, mostra a topologia utilizada no cenário 3, com identificação dos dispositivos utilizados.

Elemento	Rede	Máquina hospedeira
Zabbix	Rede de Serviços	M4
Linux	Rede de Serviços	M1
pfsense	Rede de Serviços	M6
DD-WRT	Rede Cliente	R1
pfsense	Rede Cliente	M5
Linux	Rede Cliente	M2
Windows	Rede Cliente	M3

Tabela 4.8: Elementos do Cenário 3.

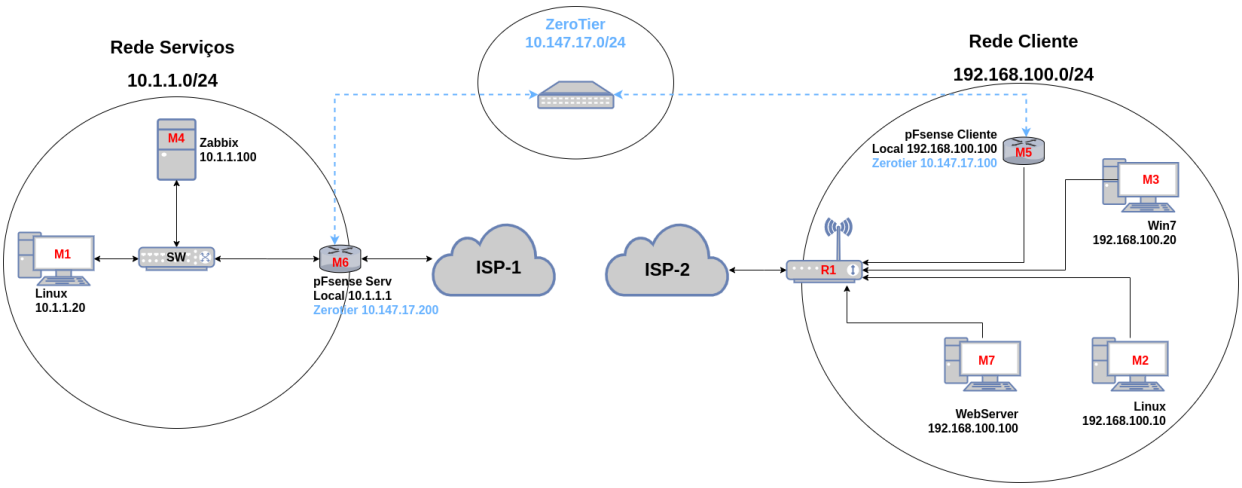


Figura 4.22: Cenário 3 Topologia - 2 redes físicas.

A figura 4.23 mostra uma foto das máquinas na rede de serviços.

Figura 4.23: Cenário 3 - Foto rede de serviços física

- figura 4.23 - detalhe A - mostra a máquina M4 onde foi instalado o Zabbix, no monitor da máquina é possível ver a tela de monitoramento de *hosts* da aplicação.
- figura 4.23 - detalhe B - mostra um *switch* utilizado para interconectar as máquinas.
- figura 4.23 - detalhe C - mostra a máquina M1 onde foi instalado o Linux da rede de serviços utilizado para administração das aplicações.
- figura 4.23 - detalhe D - mostra a máquina M6 onde foi instalado o sistema operacional pFsense da rede de serviços, responsável pela conexão com o túnel ZeroTier e Gateway da rede.

A descrição dos equipamentos físicos pode ser entrada na tabela 3.1.

A figura 4.24 mostra uma foto das máquinas que compõe a rede cliente física.

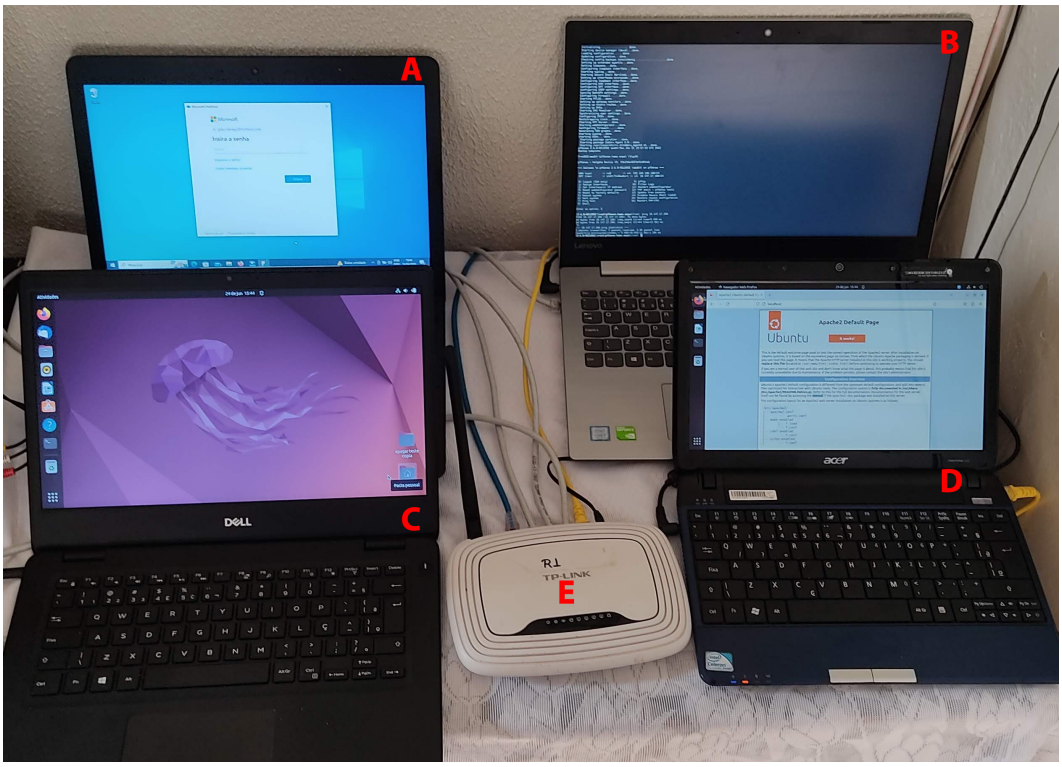


Figura 4.24: Cenário 3 - Foto rede cliente com máquinas físicas

- figura 4.24 - detalhe A - mostra a máquina M3 onde foi instalado o sistema operacional Windows.
- figura 4.24 - detalhe B - mostra a máquina M5 onde foi instalado o sistema operacional pFsense da rede cliente, responsável pela conexão com o túnel ZeroTier.
- figura 4.24 - detalhe C - mostra a máquina M2 onde foi instalado um sistema operacional Linux.

- figura 4.24 - detalhe D - mostra a máquina M7 onde foi instalado um sistema operacional Linux com um *WebServer*, na tela da máquina é possível ver a página inicial do servidor.
- figura 4.24 - detalhe E - mostra o equipamento R1 onde foi instalado o *firmware* DD-WRT.

A descrição dos equipamentos físicos pode ser entrada na tabela 3.1.

Teste de transmissão

Teste de taxa de transmissão pela *Internet* com a ferramenta SIMET. A figura 4.25 mostra a saída dos testes realizados nas máquinas das duas redes físicas, Linux da rede de serviços e Linux da rede cliente.

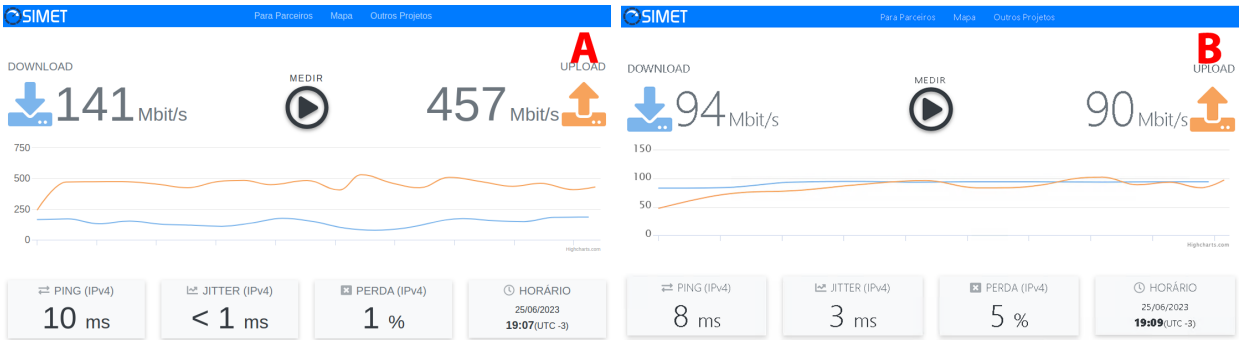


Figura 4.25: Cenário 3 - Teste de transferência com a *Internet*

- figura 4.25 - detalhe A - mostra a saída do teste realizado na máquina Linux da rede de serviços.
- figura 4.25 - detalhe B - mostra a saída do teste realizado na máquina Linux da rede de cliente.

Sobre os valores obtidos, na rede cliente, a taxa continua a ser limitada à capacidade do roteador de saída que é de até 100Mbits nominais. Na rede de serviços uma das placas de rede da máquina pFsense é um adaptador de rede USB, o que também gerou alguma limitação.

Teste de PING

Caminho dos pacotes entre as redes

O caminho traçado dos pacotes saindo da rede de serviços, passando pelo *gateway* do túnel e chegando ao destino na rede cliente, pode ser visto na figura 4.26.

```
aluno@zabbix: ~
aluno@zabbix:~$ ip -o -4 a | grep inet | cut -d" " -f1-9
1: lo    inet 127.0.0.1/8 scope host
2: enp3s0f1    inet 10.1.1.100/24 brd 10.1.1.255
4: virbr0    inet 192.168.122.1/24 brd 192.168.122.255
5: docker0    inet 172.17.0.1/16 brd 172.17.255.255
aluno@zabbix:~$
aluno@zabbix:~$ ip route
default via 10.1.1.1 dev enp3s0f1 proto dhcp metric 100
10.1.1.0/24 dev enp3s0f1 proto kernel scope link src 10.1.1.100 metric 100
169.254.0.0/16 dev virbr0 scope link metric 1000 linkdown
172.17.0.0/16 dev docker0 proto kernel scope link src 172.17.0.1 linkdown
192.168.122.0/24 dev virbr0 proto kernel scope link src 192.168.122.1 linkdown
aluno@zabbix:~$
aluno@zabbix:~$ traceroute 192.168.100.10
traceroute to 192.168.100.10 (192.168.100.10), 30 hops max, 60 byte packets
 1  pfSense.home.arpa (10.1.1.1)  0.131 ms  0.104 ms  0.052 ms
 2  10.147.17.100 (10.147.17.100)  11.604 ms  11.610 ms  11.685 ms
 3  192.168.100.10 (192.168.100.10)  11.760 ms  11.840 ms  11.918 ms
aluno@zabbix:~$
```

A

B

C

Figura 4.26: Cenário 3 Teste de traceroute da rede de serviços para a rede cliente.

Detalhes da figura 4.26:

- figura 4.26 - detalhe A - mostra as interfaces de rede da máquina Zabbix na rede de serviços, sendo a interface **enp3S0f1** a interface de saída com endereço IP 10.1.1.100/24;
- figura 4.26 - detalhe B - mostra a rota padrão da rede apontando para a máquina 10.1.1.1 que é o pfsense da rede de serviços e funciona como *gateway* para a *Internet* e para o túnel que interliga as duas redes através da *Internet*;
- figura 4.26 - detalhe C - mostra o caminho percorrido pelos pacotes da rede de serviços até a máquina Linux da rede cliente, passando pelo *gateway* na rede 10.1.1.0/24, em seguida pelo *gateway* ZeroTier na rede cliente e chegando a máquina Linux da rede cliente.

Teste de conexão

Na máquina Linux da rede de serviços, endereço IP 10.1.1.10 foram efetuados disparos de ping com 100 pacotes cada, para a máquina Linux na rede cliente com endereço IP 192.168.100.10, os disparos foram enviados a cada 15 minutos no período de 24 horas. Foi criado um pequeno *script* para fazer os disparos e salvar a saída em um arquivo de log, um agendamento no cron foi criado para executar o *scritp*.

Ping perda de pacotes - foram realizados 96 disparos de 100 pacotes cada, ainda assim, e houve apenas 1% de perda em um dos testes. O gráfico apresentado na figura 4.27 gerado com o log dos disparos, mostra o resultado dos testes.

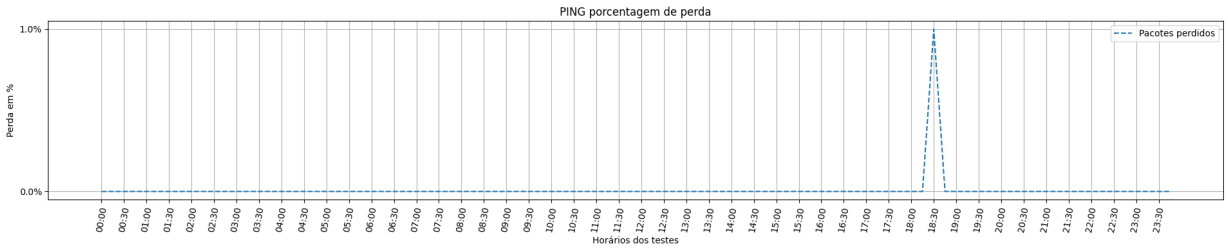


Figura 4.27: Cenário 3 Teste de PING - Perda de pacotes.

Ping RTT - foram realizados 96 disparos de 100 pacotes cada, nestes testes, o RTT médio variou de 6.4ms à 11.8ms, no ping entre a rede cliente e rede de serviços, considerando que o RTT médio da rede hospedeira para o servidor do Google na *internet* foi de 18.6ms como pode ser visto na figura 4.1, os valores de RTT obtidos nos testes podem ser considerados baixos e aceitáveis. O gráfico apresentado na figura 4.28 gerado com o log dos disparos, mostra a variação do RTT nos testes.

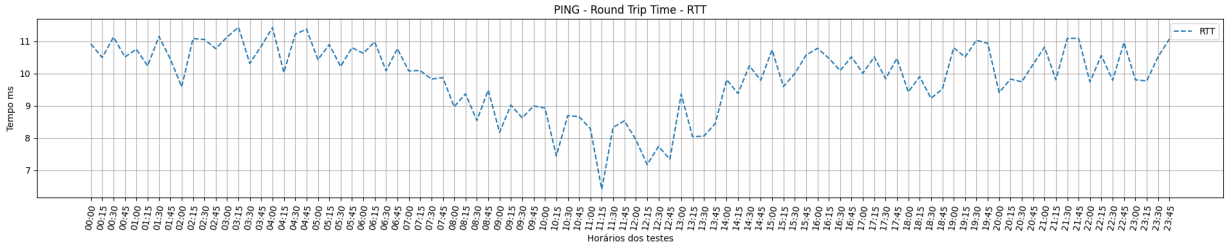


Figura 4.28: Cenário 3 Teste de PING - RTT.

Teste com iPerf

Nos testes com iPerf3, a máquina Linux na rede cliente funcionou como servidor e a máquina Linux na rede de serviços funcionou como cliente, enviado os dados. Foram feitos 96 testes de transmissão com configurações padrão do iPerf3 e sem especificação de tamanho de arquivo. O gráfico presente na figura 4.29 mostra a variação das taxas de transmissão obtidas nos testes.

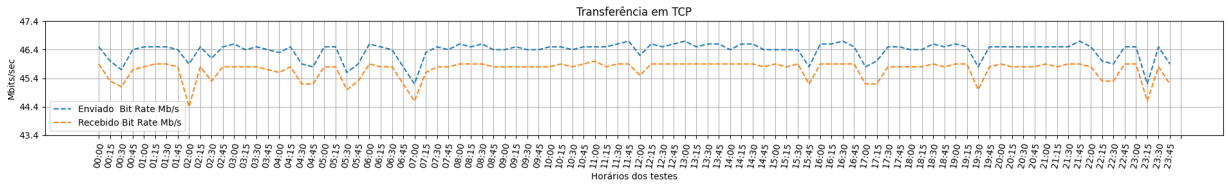


Figura 4.29: Cenário 3 Teste iPerf3 - Taxa de transmissão.

O gráfico na figura 4.30, mostra as taxas de retransmissão de pacotes obtidas durante os testes.

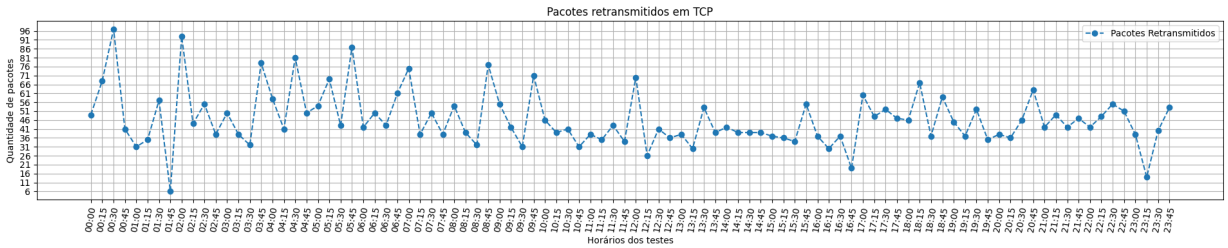


Figura 4.30: Cenário 3 Teste iPerf3 - Pacotes retransmitidos.

A tabela 4.9 mostra dos resultados obtidos:

Bandwidth Mbits/sec		**Retransmitions**
Sender	Receiver	Packets
46.3656	45.6968	46

Tabela 4.9: Cenário 3 médias das taxas iPerf3.

Mesmo considerando as limitações do adaptador de rede USB e do roteador de borda citadas anteriormente, os níveis de transmissão com iPerf3 obtidos foram baixos, porém para os objetivos de acesso remoto e monitoramento, não significam um impedimento, enquanto a quantidade de pacotes retransmitidos se manteve baixa.

Teste com Rsync

No teste com rsync a máquina Zabbix da rede de serviços funcionou como origem e a máquina Linux da rede cliente como destino, foram efetuados três testes de transferência, com arquivos de 100Mbytes. Seguem os comandos:

```
#Maquina Linux rede de cliente como destino dos dados
rync -avhP arquivo_100M aluno@192.168.100.10:/home/aluno
```

A figura 4.31 mostra os resultados dos testes.

```
aluno@zabbix:~$ rsync -avhP arquivo_100M aluno@192.168.100.10:/home/aluno
aluno@192.168.100.10's password:
sending incremental file list
arquivo_100M
        104,86M 100%    5,56MB/s    0:00:17 (xfr#1, to-chk=0/1)

sent 104,88M bytes  received 35 bytes  4,88M bytes/sec
total size is 104,86M  speedup is 1,00
aluno@zabbix: $
aluno@zabbix:~$ rsync -avhP arquivo_100M aluno@192.168.100.10:/home/aluno
aluno@192.168.100.10's password:
sending incremental file list
arquivo_100M
        104,86M 100%    5,53MB/s    0:00:18 (xfr#1, to-chk=0/1)

sent 104,88M bytes  received 35 bytes  4,88M bytes/sec
total size is 104,86M  speedup is 1,00
aluno@zabbix: $
aluno@zabbix:~$ rsync -avhP arquivo_100M aluno@192.168.100.10:/home/aluno
aluno@192.168.100.10's password:
sending incremental file list
arquivo_100M
        104,86M 100%    5,61MB/s    0:00:17 (xfr#1, to-chk=0/1)

sent 104,88M bytes  received 35 bytes  4,88M bytes/sec
total size is 104,86M  speedup is 1,00
aluno@zabbix: $
```

Figura 4.31: Cenário 3 saída transferência com rsync.

A tabela 4.10 mostra as médias dos resultados obtidos:

Taxa de transmissão		
Mínimo	Média	Máximo
4.88 MBytes/sec	4.88 MBytes/sec	4.88 MBytes/sec

Tabela 4.10: Cenário 3 resultados rsync.

Os três testes com rsync obtiveram o mesmo valor de taxa de transferência média, o que pode significar estabilidade, apesar de um pouco baixas, as taxas continuam dentro do esperado para uma rede com largura de banda nominal de 100Mbits, sendo o caso da rede cliente.

Conexão com Zabbix

A figura 4.32 mostra o painel de *hosts* monitorados pelo Zabbix, como pode ser observado na coluna *Availability* as máquinas monitoras estão *on-line*, neste cenário, são monitoradas as máquinas Linux, Windows, WebServer e pfSense na rede cliente.

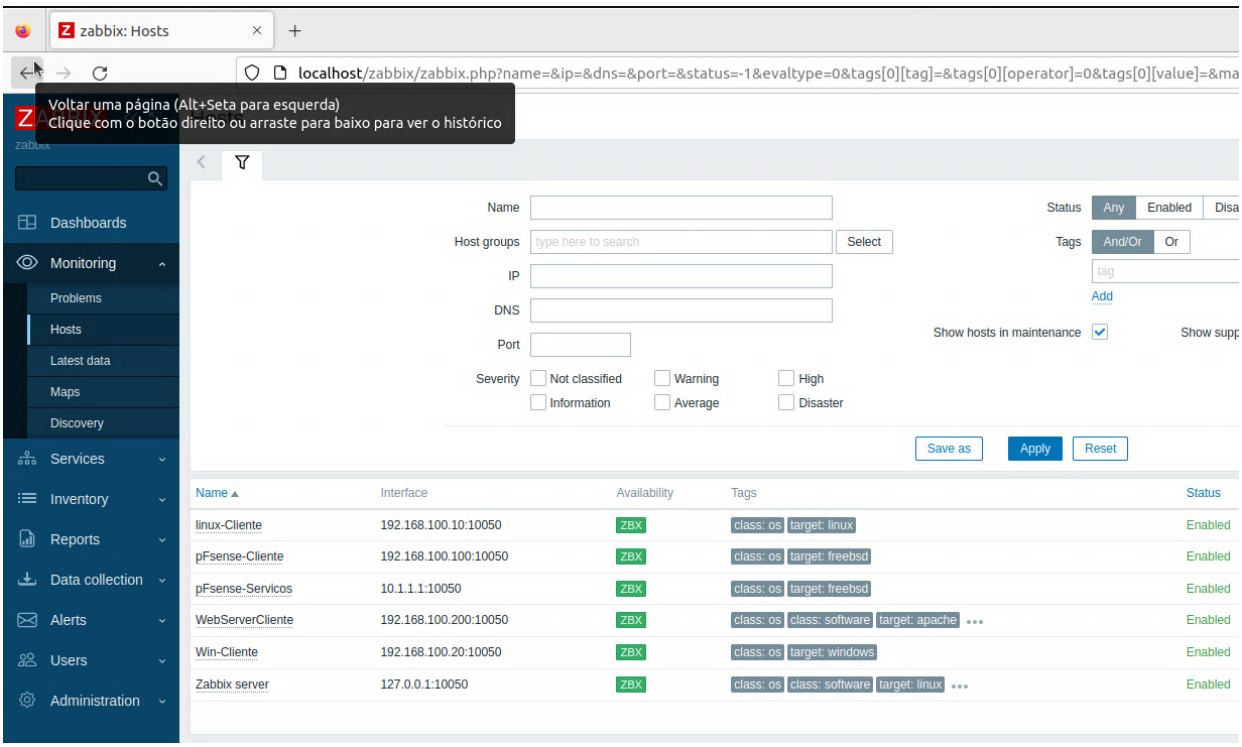


Figura 4.32: Cenário 3 tela de monitoramento Zabbix.

Acesso remoto

Durante os testes com rsync também é testado o acesso remoto via SSH entre as máquinas Linux, pois o rsync usa o SSH para estabelecer conexão.

O teste de acesso remoto a máquina Windows foi realizado com a aplicação Remmina e o protocolo RDP. A figura 4.33 mostra a tela da aplicação Remmina durante o acesso remoto à máquina Windows.

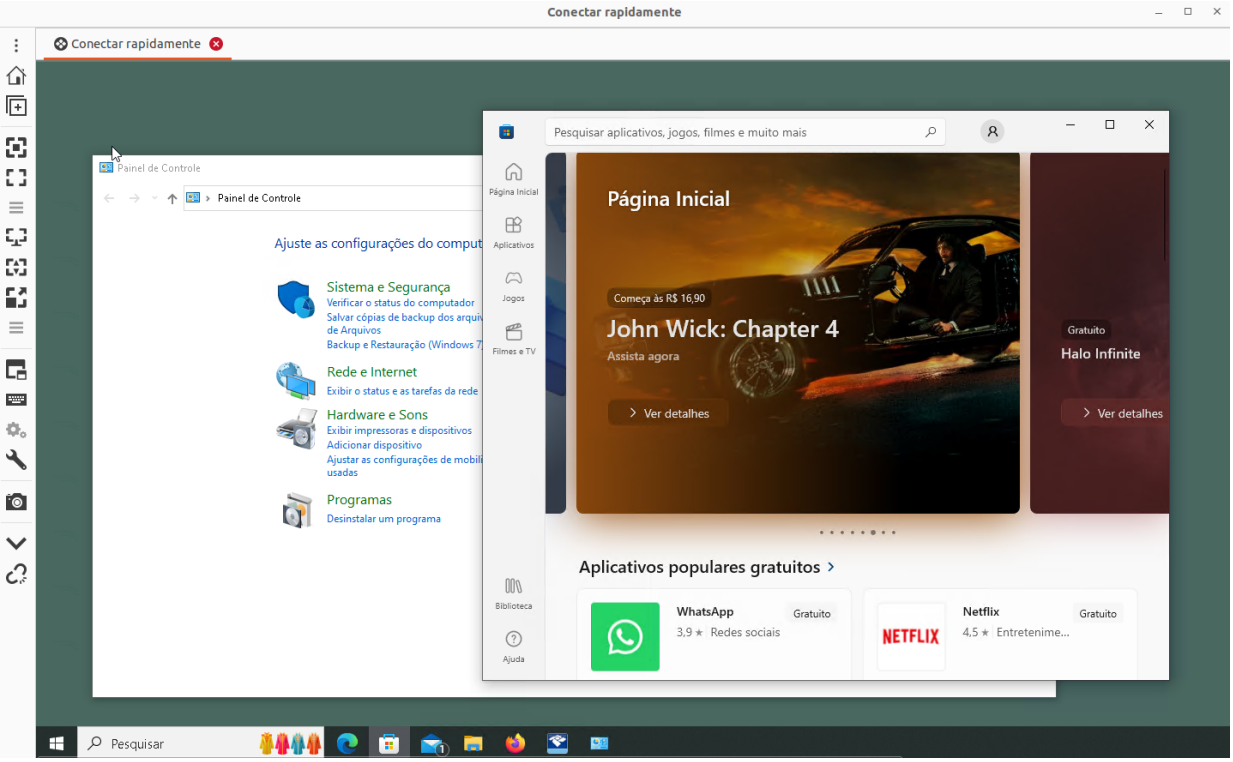


Figura 4.33: Cenário 3 acesso remoto Windows.

Os resultados obtidos nos testes do cenário 3, totalmente físico, mostram estabilidade nas conexões e, apesar das taxas de transferências mais baixas que no cenário 2, os testes ainda indicam que ser possível monitorar e gerenciar a rede cliente remotamente.

4.2.5 Cenário 4

Cenário 4 - virtualizado em 2 máquinas diferentes - a rede de serviços foi virtualizada na máquina física M4 e a rede cliente foi virtualizada na máquina física M1. Neste cenário, as redes foram virtualizadas em 2 máquinas físicas diferentes, visando aproveitar melhor os recursos das máquinas hospedeiras, possibilitando o aumento dos recursos de memória de processador reservados para algumas máquinas virtuais nas redes. A tabela 4.11 apresenta os elementos em cada rede do cenário, sua respectiva máquina hospedeira e a especificação dos recursos reservados para cada elemento.

A figura 4.34, mostra a topologia utilizada no cenário 4, com identificação dos dispositivos utilizados.

Elemento	Memória	Processador	Máquina hope-deira	Rede
DD-WRT v3.0	1024 MB	2	M1	Rede cliente
pFsense2.6	2048 MB	2	M1	Rede cliente
Ubuntu Server	1024 MB	1	M1	Rede cliente
Windows7	1024 MB	1	M1	Rede cliente
Xubuntu	2048 MB	2	M1	Rede cliente
Kali Linux 2023.2	1024 MB	1	M1	Rede cliente
pFsense2.6	4096 MB	2	M4	Rede de Serviços
Wazuh	4096 MB	2	M4	Rede de Serviços
Xubuntu	4096 MB	2	M4	Rede de Serviços
Zabbix5	2048 MB	2	M4	Rede de Serviços

Tabela 4.11: Máquinas elementos do cenário 4.

Figura 4.34: Cenário 4 Topologia - 2 redes virtuais em 2 máquinas físicas diferentes.

A figura 4.35 mostra uma foto das máquinas na rede de serviços.

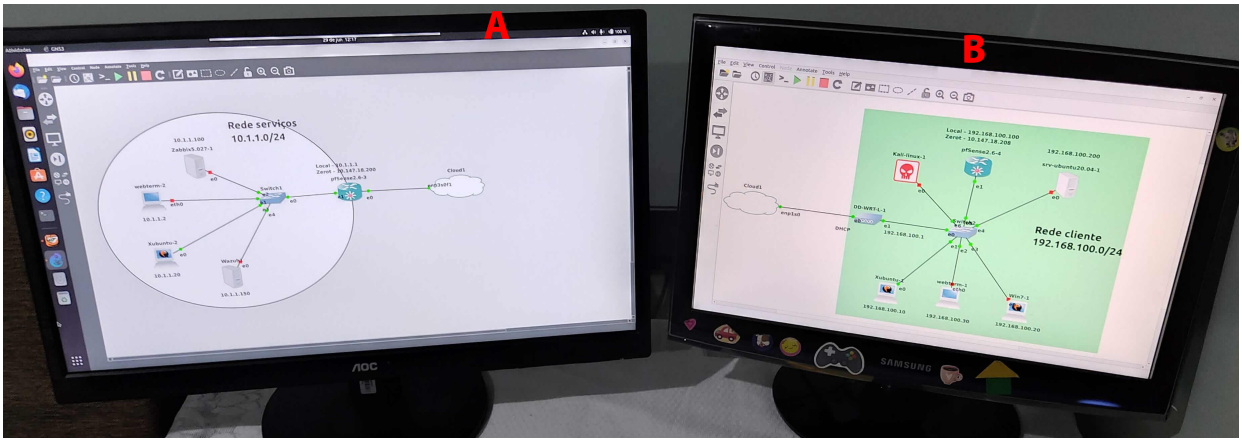


Figura 4.35: Cenário 4 - Tela das redes virtualizadas

- figura 4.35 - detalhe A - a tela da máquina M4 onde foi virtualizada a rede de serviços.

- figura 4.35 - detalhe B - a tela da máquina M1 onde foi virtualizada a rede cliente.

A descrição dos equipamentos físicos pode ser entrada na tabela 3.1.

Teste de transmissão

Teste de taxa de transmissão pela *Internet* com a ferramenta SIMET. A figura 4.36 mostra a saída dos testes realizados nas máquinas das duas redes virtuais.

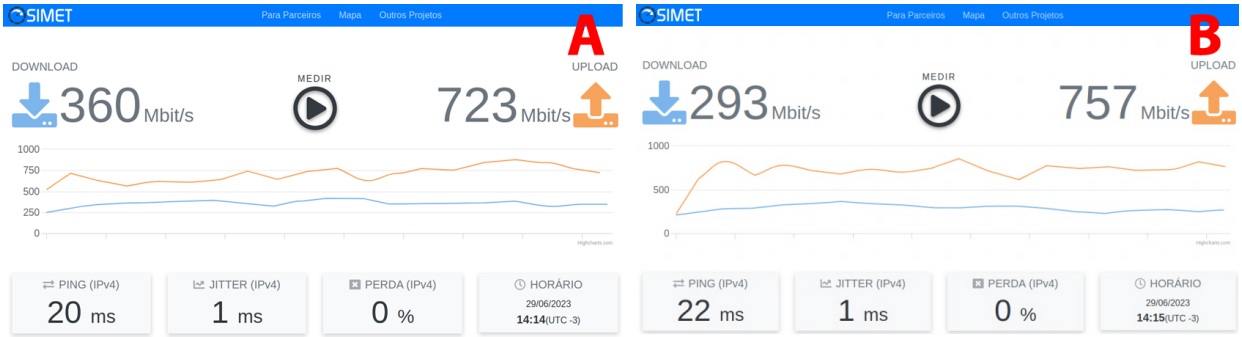


Figura 4.36: Cenário 4 - Teste de transferência com a *Internet*

- figura 4.36 - detalhe A - mostra a saída do teste realizado na máquina Linux da rede de serviços.
- figura 4.36 - detalhe B - mostra a saída do teste realizado na máquina Linux da rede de cliente.

Teste de PING

Caminho dos pacotes entre as redes

O caminho traçado dos pacotes saindo da rede de serviços, passando pelo *gateway* do túnel e chegando ao destino na rede cliente, pode ser visto na figura 4.37.

```
Terminal - aluno@xubuntu: ~
Arquivo  Editar  Ver  Terminal  Abas  Ajuda
aluno@xubuntu:~$ ip -o -4 a | grep inet | cut -d" " -f1-9
1: lo    inet 127.0.0.1/8 scope host
2: ens4    inet 10.1.1.20/24 brd 10.1.1.255
aluno@xubuntu:~$
aluno@xubuntu:~$ ip route
default via 10.1.1.1 dev ens4 proto static
10.1.1.0/24 dev ens4 proto kernel scope link src 10.1.1.20
aluno@xubuntu:~$
aluno@xubuntu:~$ traceroute 192.168.100.10
traceroute to 192.168.100.10 (192.168.100.10), 30 hops max, 60 byte packets
 1  _gateway (10.1.1.1)  2.739 ms  2.511 ms  2.370 ms
 2  10.147.18.208 (10.147.18.208)  9.034 ms  9.084 ms  8.839 ms
 3  192.168.100.10 (192.168.100.10)  9.136 ms  9.164 ms  9.113 ms
aluno@xubuntu:~$
```

A

B

C

Figura 4.37: Cenário 4 Teste de traceroute da rede de serviços para a rede cliente.

Detalhes da figura 4.37:

- figura 4.37 - detalhe A - mostra as interfaces de rede da máquina Linux na rede de serviços, sendo a interface **ens4** a interface de saída com endereço IP 10.1.1.20/24;
- figura 4.26 - detalhe B - mostra a rota padrão da rede apontando para a máquina 10.1.1.1 que é o pfsense da rede de serviços e funciona como *gateway* para a *Internet* e para o túnel que interliga as duas redes através da *Internet*;
- figura 4.37 - detalhe C - mostra o caminho percorrido pelos pacotes da rede de serviços até a máquina Linux da rede cliente, passando pelo *gateway* na rede 10.1.1.0/24, em seguida pelo *gateway* ZeroTier na rede cliente endereço IP 10.147.18.208 e chegando a máquina Linux da rede cliente.

Teste de conexão

Na máquina Linux da rede de serviços, endereço IP 10.1.1.10 foram efetuados disparos de ping com 100 pacotes cada, para a máquina Linux na rede cliente com endereço IP 192.168.100.10, os disparos foram enviados a cada 15 minutos no período de 24 horas. Foi criado um pequeno *script* para fazer os disparos e salvar a saída em um arquivo de log, um agendamento no cron foi criado para executar o *scritp*.

Ping perda de pacotes - foram realizados 96 disparos de 100 pacotes cada, com 0% de perda nos testes. O gráfico apresentado na figura 4.38 gerado com o log dos disparos, mostra o resultado dos testes.

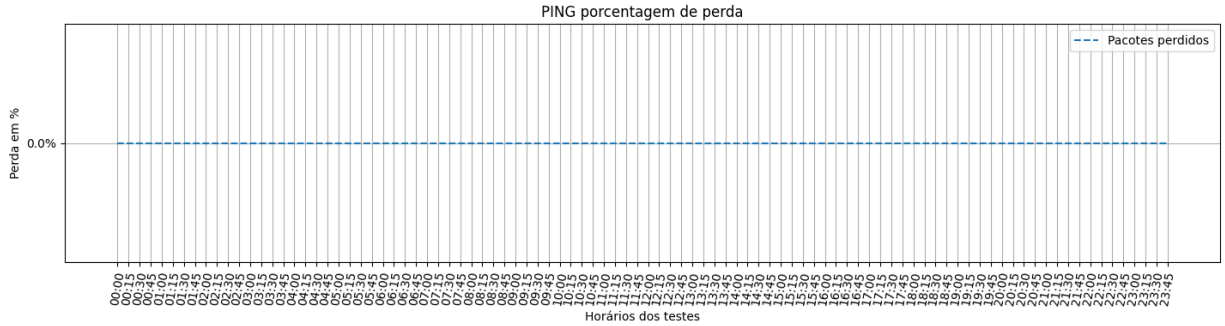


Figura 4.38: Cenário 4 Teste de PING - Perda de pacotes.

Ping RTT - foram realizados 96 disparos de 100 pacotes cada, nestes testes, o RTT médio variou de 4.06ms à 6.40ms, no ping entre a rede cliente e rede de serviços, considerando que o RTT médio da rede hospedeira para o servidor do Google na *internet* foi de 18.6ms como pode ser visto na figura 4.1, os valores de RTT obtidos nos testes podem ser considerados baixos. O gráfico presentado na figura 4.39 gerado com o log dos disparos, mostra a variação do RTT nos testes.

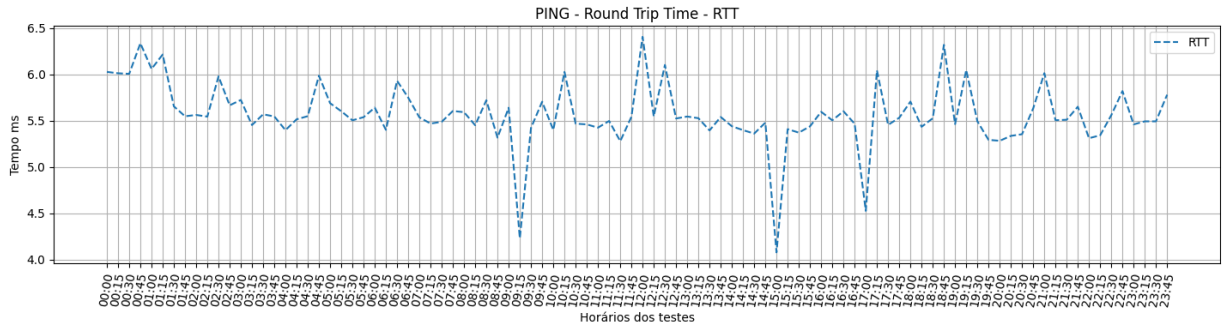


Figura 4.39: Cenário 4 Teste de PING - RTT.

Teste com iPerf

Nos testes com iPerf3, a máquina Linux na rede cliente funcionou como servidor e a máquina Linux na rede de serviços funcionou como cliente, enviado os dados. Foram feitos 96 testes de transmissão com configurações padrão do iPerf3 e sem especificação de tamanho de arquivo. O gráfico presente na figura 4.40 mostra a variação das taxas de transmissão obtidas nos testes.

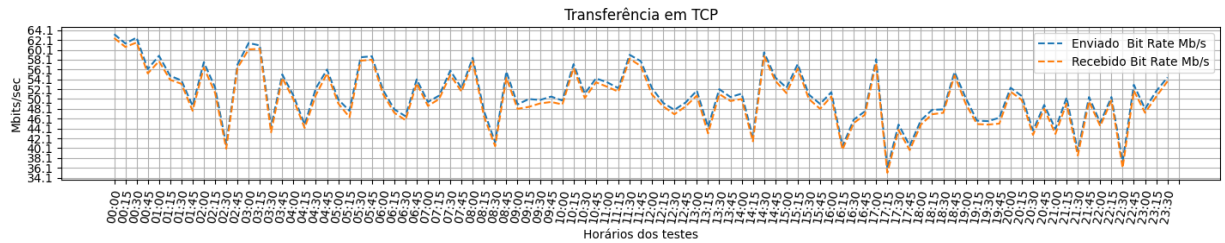


Figura 4.40: Cenário 4 Teste iPerf3 - Taxa de transmissão.

O gráfico na figura 4.41, mostra as taxas de retransmissão de pacotes obtidas durante os testes.

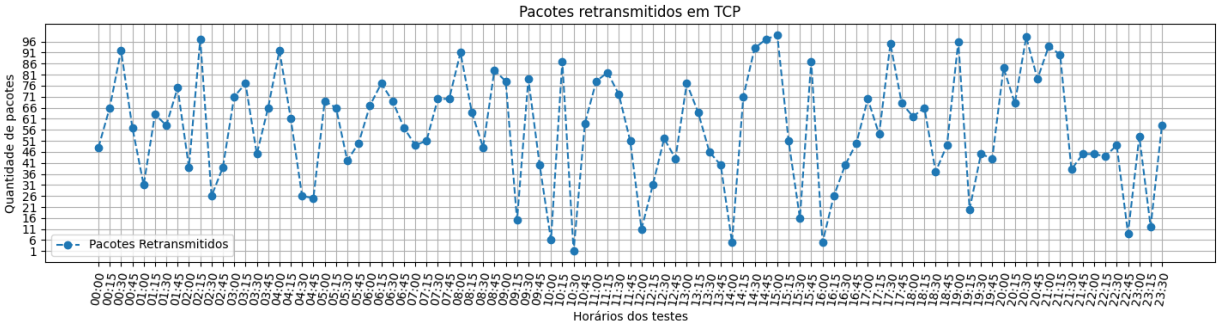


Figura 4.41: Cenário 4 Teste iPerf3 - Pacotes retransmitidos.

A tabela 4.12 mostra dos resultados obtidos:

Bandwidth Mbits/sec		**Retransmitions**
Sender	Receiver	Packets
50.9684	50.0694	56

Tabela 4.12: Cenário 4 médias das taxas iPerf3.

Os níveis de transmissão com iPerf3 obtidos foram baixos quando comparados com a taxa de transmissão com a internet, porém para os objetivos de acesso remoto e monitoramento, não significam um impedimento, enquanto a quantidade de pacotes retransmitidos se manteve baixa.

Teste com Rsync

No teste com rsync a máquina de Linux da rede de serviços funcionou como origem e a máquina Linux da rede cliente como destino, foram efetuados três testes de transferência, com arquivos de 100Mbytes. Seguem os comandos:

```
#Maquina Linux rede de cliente como destino dos dados
rync -avhP arquivo_100M aluno@192.168.100.10:/home/aluno
```

A figura 4.42 mostra os resultados dos testes.

```
aluno@xubuntu:~$ rsync -avhP arquivo_100M 192.168.100.10:/home/aluno
aluno@192.168.100.10's password:
sending incremental file list
arquivo_100M
        104.86M 100%    5.76MB/s    0:00:17 (xfr#1, to-chk=0/1)

sent 104.88M bytes  received 35 bytes  4.88M bytes/sec
total size is 104.86M  speedup is 1.00
aluno@xubuntu:~$
aluno@xubuntu:~$ rsync -avhP arquivo_100M 192.168.100.10:/home/aluno
aluno@192.168.100.10's password:
sending incremental file list
arquivo_100M
        104.86M 100%    4.69MB/s    0:00:21 (xfr#1, to-chk=0/1)

sent 104.88M bytes  received 35 bytes  4.28M bytes/sec
total size is 104.86M  speedup is 1.00
aluno@xubuntu:~$
aluno@xubuntu:~$ rsync -avhP arquivo_100M 192.168.100.10:/home/aluno
aluno@192.168.100.10's password:
sending incremental file list
arquivo_100M
        104.86M 100%    5.81MB/s    0:00:17 (xfr#1, to-chk=0/1)

sent 104.88M bytes  received 35 bytes  5.12M bytes/sec
total size is 104.86M  speedup is 1.00
aluno@xubuntu:~$
```

Figura 4.42: Cenário 4 saída transferência com rsync.

A tabela 4.13 mostra as médias dos resultados obtidos:

Taxa de transmissão		
Mínimo	Média	Máximo
4.28 MBytes/sec	4.76 MBytes/sec	5.12 MBytes/sec

Tabela 4.13: Cenário 4 resultados rsync.

Os três testes com rsync obtiveram valores de taxa de transferência média próximos e sem erros, demonstrando estabilidade na conexão entre as redes.

Conexão com Zabbix

A figura 4.43 mostra o painel de *hosts* monitorados pelo Zabbix, na coluna *Availability* pode ser observado o estado das máquinas monitoras. Neste cenário, são monitoradas as máquinas da rede cliente e também as máquinas da rede de serviços. Ainda nesta figura, é possível notar que a máquina Windows está *offline*, foi criado um bot integrando o Zabbix ao messageiro Telegram para o envio de mensagens de alerta, a figura 4.44 mostra o alerta gerado via aplicativo de mensagens, informando sobre o estado da máquina Windows, primeiro como inacessível, pois a máquina estava desligada e em seguida novamente acessível, após a máquina ser religada.

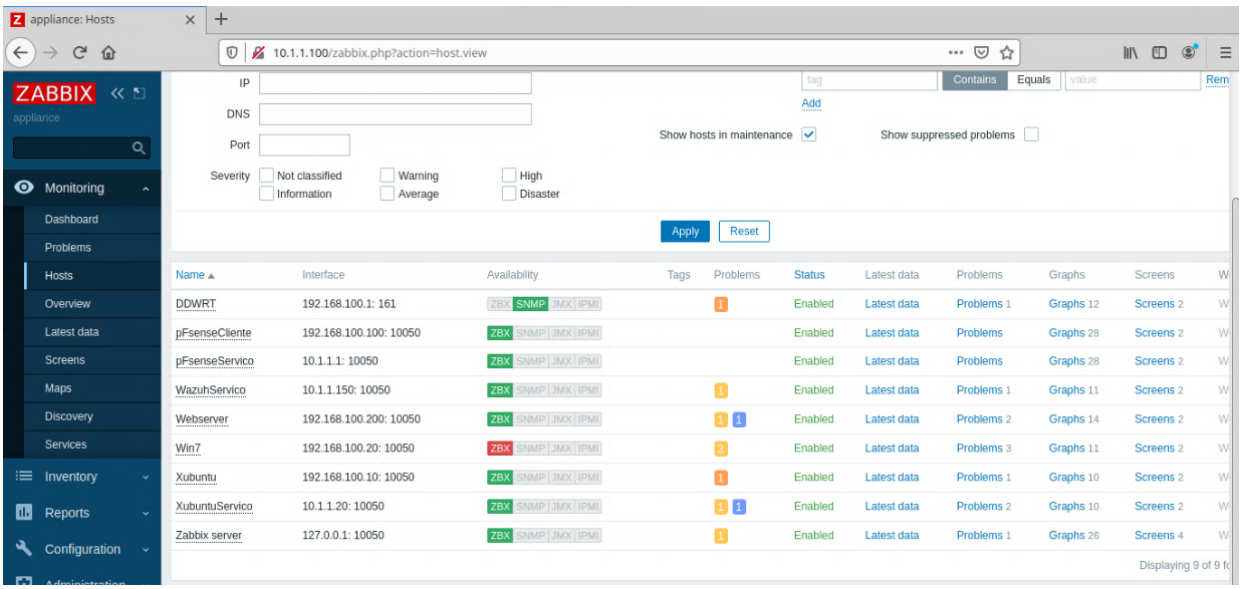


Figura 4.43: Cenário 4 tela de monitoramento Zabbix.

Figura 4.44: Cenário 4 tela alerta de estado Zabbix/Telegram.

Como demonstrado anteriormente, o Zabbix também permite criar mapas dinâmicos dos dispositivos monitorados, onde é possível visualizar o estado dos dispositivos, erros e aletas, estado dos enlaces com taxas de entrada e saída. Neste cenário, foi criado um mapa com as duas redes, onde até mesmo o enlace do túnel VPN entre as redes está sendo monitorado. O mapa pode ser visto na figura 4.45

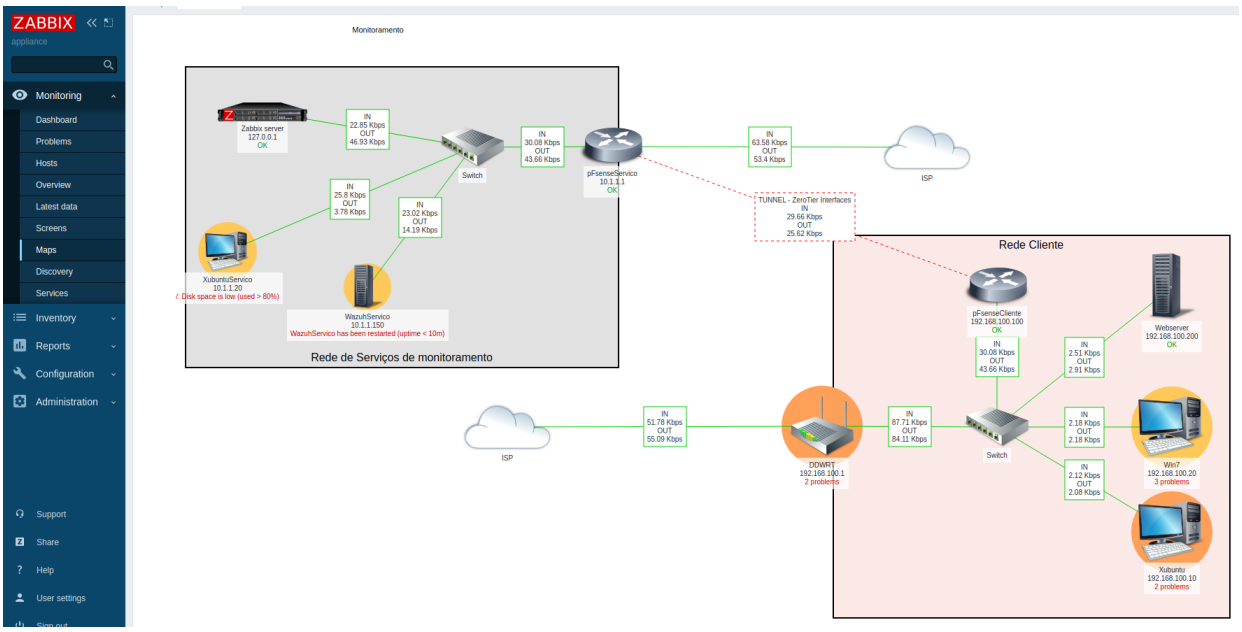


Figura 4.45: Cenário 4 Mapa de rede interativo Zabbix.

Testes de segurança

Para testar a capacidade de detecção de ameaças e emissão de alertas das soluções implantadas, alguns testes de segurança foram executados. Para isso, uma máquina atacante foi adiciona ao cenário na rede cliente como pode ser visto na descrição do cenário e na figura 4.34.

O Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil **CERT.br** é um grupo de respostas a incidentes mantido pelo NIC.br, em sua página é possível acessar algumas estatísticas sobre incidentes relatados ao grupo. Na figura 4.46 pode-se observar um gráfico de incidentes notificados de janeiro a julho de 2023, divididos por categorias, onde pode ser notado que a **varredura de portas ou *scan port*** é o incidente mais frequente, seguido pela soma de outros incidentes não nomeados e com ataques de ***Denial of Service* DoS** em terceiro lugar entre os incidentes mais relatados.

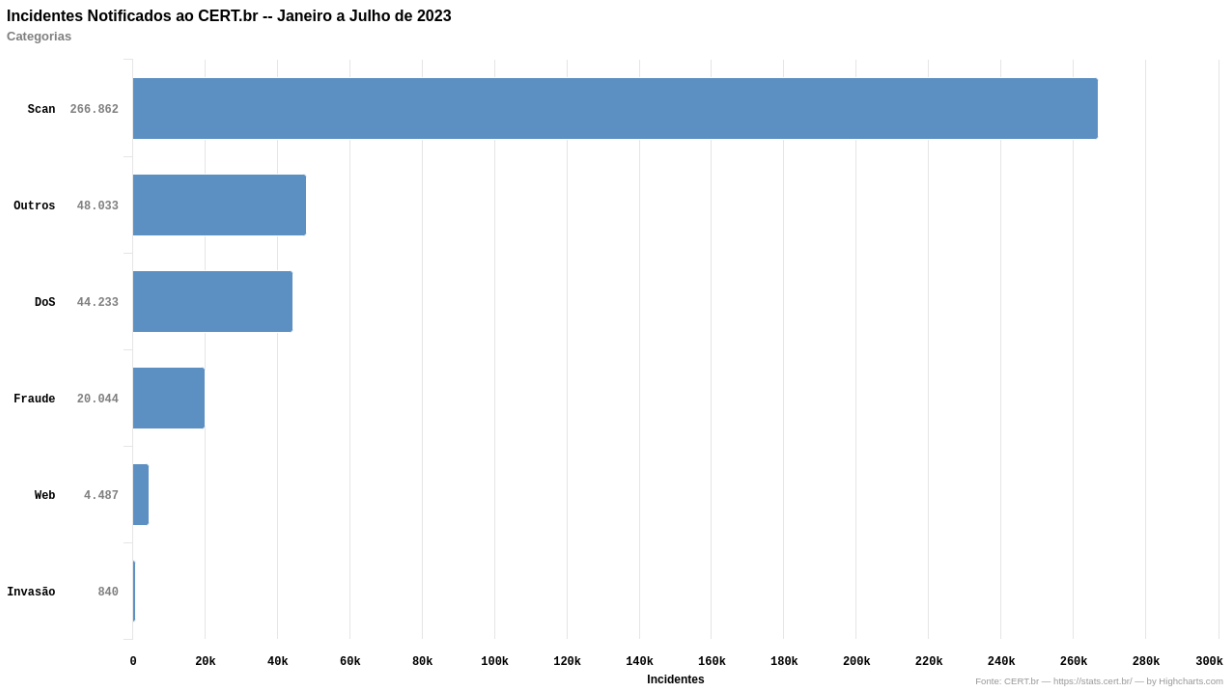


Figura 4.46: Estatística CERT.br, ataques por categoria. Fonte:https://stats.cert.br/incidentes/tipos-incidente

Varredura da rede

A partir da máquina atacante foi realizada uma varredura na rede. A varredura por parte de atacantes tem o objetivo de realizar o levantamento das máquinas ativas na rede, dos serviços e portas aguardando conexão, assim como as versões dos sistemas operacionais e das aplicações, de posse dessas informações o atacante pode direcionar seus esforços em explorar vulnerabilidades específicas.

Com a ferramenta nmap é possível realizar a varredura na rede, a aplicação suporta diferentes parâmetros que possibilitam diferentes tipos de testes.

A varredura foi realizada na rede cliente com endereço 192.168.100.0/24 que abrange toda da rede, com objetivo de descobrir quais máquinas estão ativas, o comando utilizado pode ser visto abaixo.

```
nmap -A 192.168.100.0/24
```

Ainda durante a execução da varredura, a aplicação Wazuh identificou a tentativa de acesso aos serviços instalados nas máquinas monitoradas, a figura 4.47 mostra alguns dos alertas enviados via Telegram. Mostrando que a varredura foi detectada e os administradores da rede foram alertados.

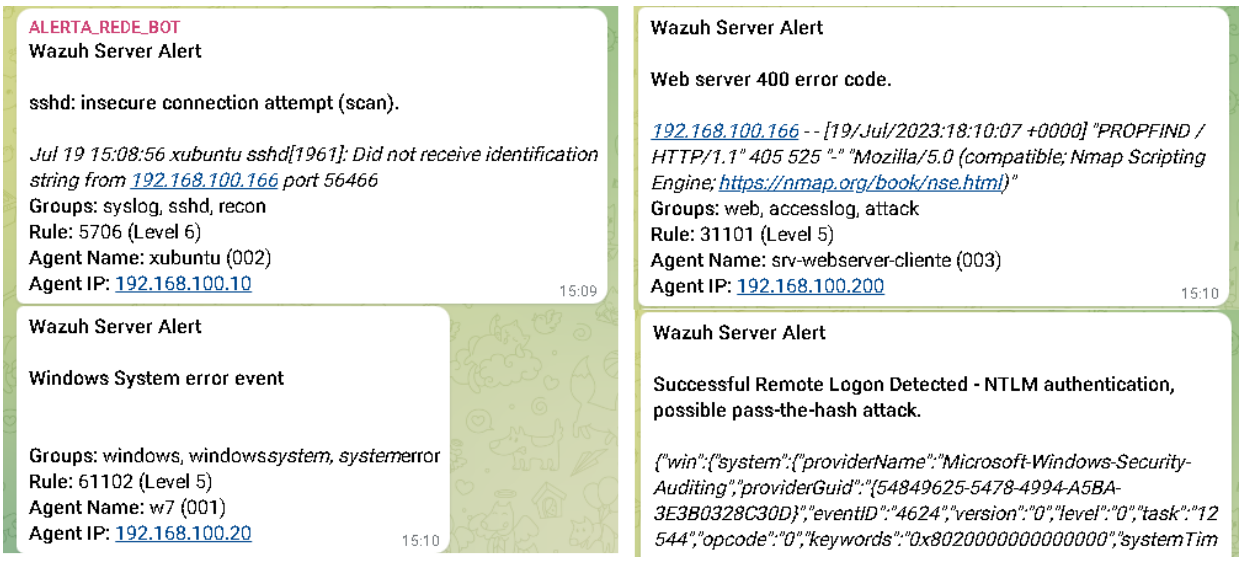
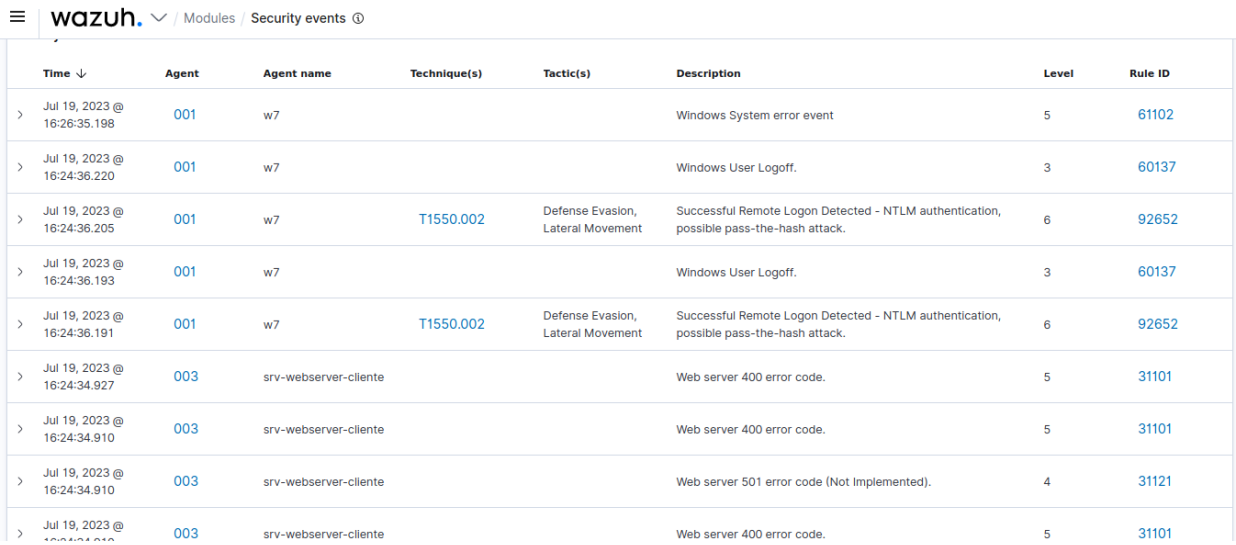


Figura 4.47: Cenário 4 alertas de varredura nmap.

No painel de eventos de segurança da aplicação é possível ver os alertas de segurança gerados, sendo possível selecionar um determinado alerta e visualizar com maiores detalhes. A figura 4.48 mostra a tela de eventos de segurança do Wazuh. A saída completa da varredura com nmap pode ser vista nos apêndices deste trabalho subseção 6.2.2.



Time	Agent	Agent name	Technique(s)	Tactic(s)	Description	Level	Rule ID
Jul 19, 2023 @ 16:26:35.198	001	w7			Windows System error event	5	61102
Jul 19, 2023 @ 16:24:36.220	001	w7			Windows User Logoff.	3	60137
Jul 19, 2023 @ 16:24:36.205	001	w7	T1550.002	Defense Evasion, Lateral Movement	Successful Remote Logon Detected - NTLM authentication, possible pass-the-hash attack.	6	92652
Jul 19, 2023 @ 16:24:36.193	001	w7			Windows User Logoff.	3	60137
Jul 19, 2023 @ 16:24:36.191	001	w7	T1550.002	Defense Evasion, Lateral Movement	Successful Remote Logon Detected - NTLM authentication, possible pass-the-hash attack.	6	92652
Jul 19, 2023 @ 16:24:34.927	003	srv-webserver-cliente			Web server 400 error code.	5	31101
Jul 19, 2023 @ 16:24:34.910	003	srv-webserver-cliente			Web server 400 error code.	5	31101
Jul 19, 2023 @ 16:24:34.910	003	srv-webserver-cliente			Web server 501 error code (Not Implemented).	4	31121
Jul 19, 2023 @	003	srv-webserver-cliente			Web server 400 error code.	5	31101

Figura 4.48: Cenário 4 tela de eventos de segurança Wazuh.

Teste de negação de serviços

O ataque de negação de serviços ou do inglês *Denial of Service* DoS, consiste em realizar diversas requisições indevidas a um servidor, a ponto de ocupar tanto os recursos da máquina, que em certo ponto ela não será mais capaz de responder a requisições dos usuários legítimos. Causando assim a indisponibilidade do serviço. Ataques convencionais de DoS inundam o servidor com uma gigantesca quantia de requisições e requerem do atacante uma abundância de recursos.

O ataque de DoS chamado *slowloris* se difere dos ataques convencionais por utilizar requisições em intervalos mais longos e regulares ocupando os recursos do servidor sem exigir muito do atacante (AKAMAI-TECHNOLOGIES, 2023). Neste trabalho o ataque de DoS foi realizado com a ferramenta *slowloris* (YALTIRAKLI, 2015).

O ataque foi disparado a partir da máquina atacante tendo como alvo o servidor *Web* da rede cliente, como pode ser visto na figura 4.49.

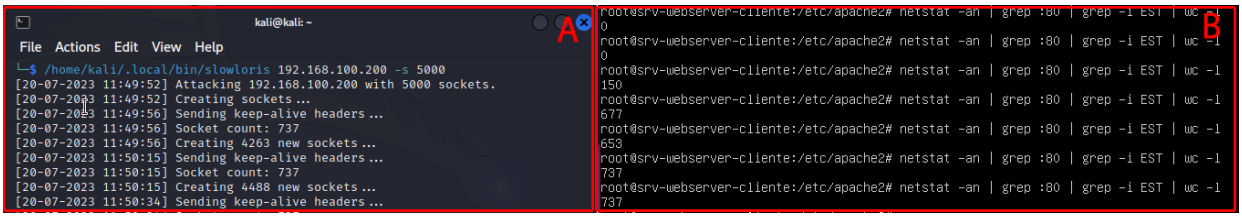


Figura 4.49: Cenário 4 Teste de ataque de negação de serviço.

Detalhes da figura 4.49:

- figura 4.49 - detalhe A - mostra o comando utilizado para disparar o ataque a partir da máquina Kali Linux, e a criação das requisições de conexão e dos quadros *keep-alive* que mantêm as conexões ocupadas;
- figura 4.49 - detalhe B - mostra no servidor *Web*, o número de conexões estabelecidas com a porta 80 aumentando.

Durante a execução do ataque a página *Web* do servidor, apesar de simples, ficou inacessível, enquanto a ferramenta Wazuh detectou a grande quantidade de requisições feitas ao servidor pelo mesmo cliente, como pode ser visto na figura 4.50.

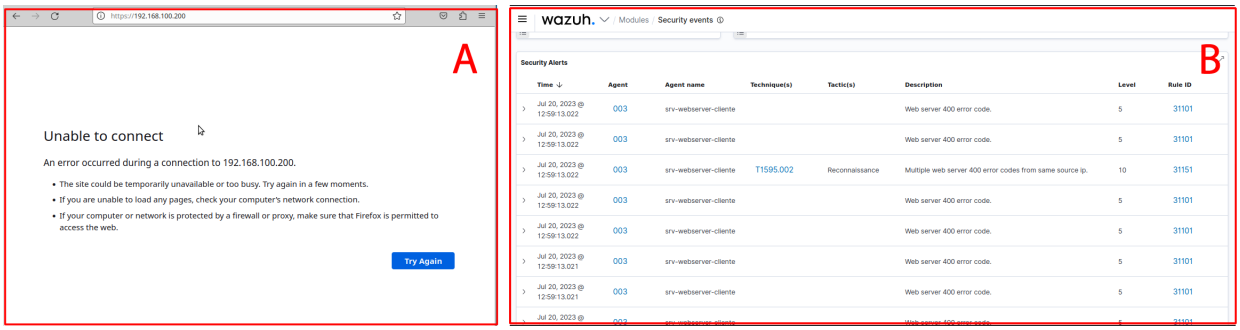


Figura 4.50: Cenário 4 Teste de ataque de negação de serviço, falha na página e detecção.

Detalhes da figura 4.50:

- figura 4.50 - detalhe A - mostra o erro no navegador ao solicitar a página do servidor que está sofrendo o ataque;
- figura 4.50 - detalhe B - mostra a tela de eventos de segurança do Wazuh ao detectar o ataque, múltiplas requisições com erro vindas do mesmo endereço IP.

A figura 4.51 mostra os alertas gerados pelo Wazuh via aplicativo de mensagem. É possível observar na imagem que um mesmo cliente está fazendo requisições diferentes ao servidor, e que a ferramenta classificou esse comportamento como ataque.

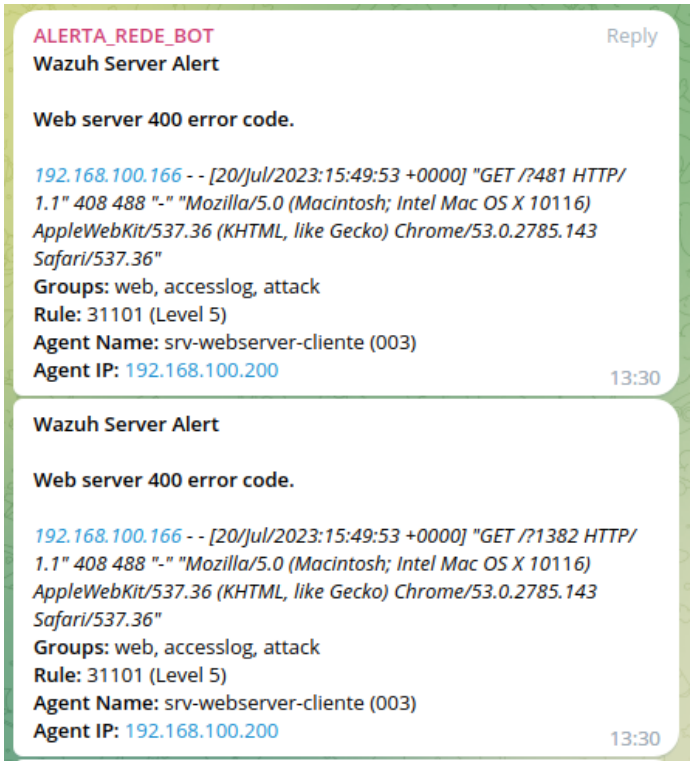


Figura 4.51: Cenário 4 Teste de ataque de negação de serviço, falha na página e detecção.

4.3 ANÁLISE DOS RESULTADOS

Nesta seção foi realizado um apanhado dos resultados obtidos na seção anterior, a fim de proporcionar melhor entendimento.

4.3.1 Testes de conexão e transmissão

Nos 4 cenários utilizados, foram realizados testes de conexão com a ferramenta PING e testes de transferência de dados com as ferramentas iPerf e Rsync.

Resumo dos testes de conexão com PING

A tabela 4.14 traz os resultados obtidos nos testes de PING. Pelos resultados é possível perceber que o cenário 3, composto totalmente por máquinas físicas, obteve os piores resultados, o que já era esperado, por se tratar de um cenário mais realista e com mais variáveis de interferência e com um ambiente menos controlado, porém mais próximo da realidade. Ainda assim, considerando que um ping para o DNS do google pode chegar a um RTT de 18ms como pode ser visto na subseção 4.2.1, o RTT de 11ms obtido neste teste pode ser considerado um bom resultado.

Teste de conexão – PING	Perda	RTT avg ms
Cenário 1 – duas redes virtualizadas na mesma máquina	0,00%	2,58
Cenário 2 – cenário misto, uma rede física e uma rede virtualizada	0,00%	2,80
Cenário 3 – totalmente físico	1,00%	11,80
Cenário 4 – virtualizado em 2 máquinas diferentes	0,00%	6,40

Tabela 4.14: Resumo resultados dos testes de PING.

A figura 4.52 apresenta os resultados dos testes de PING em forma de gráfico.

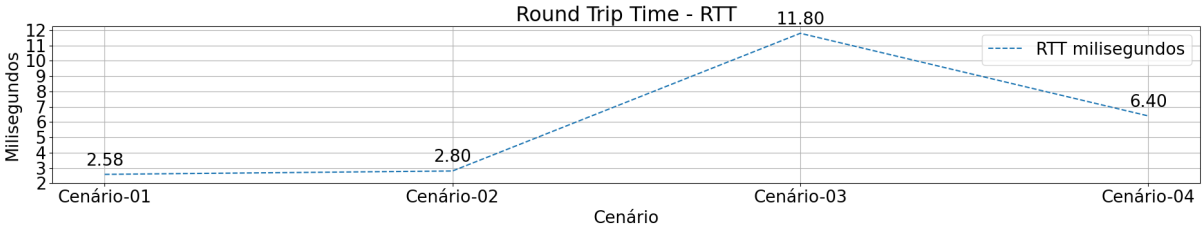


Figura 4.52: Resumo do resultado dos testes de PING.

Resumo dos testes de transmissão com iPerf

A tabela 4.15 traz os resultados obtidos nos testes de transmissão de dados com a ferramenta iPerf. Pelos resultados obtidos é possível notar que, o cenário 1 foi onde ocorreram os piores resultados, e que a diferença do cenário 3 para os cenários 2 e 4 não foi grande.

Teste transmissão iPERF	Sender Mbits/sec	Receiver Mbits/sec	Retransmitted packets avg
Cenário 1 – duas redes virtualizadas na mesma máquina	18,87	18,6	18
Cenário 2 – cenário misto, uma rede física e uma rede virtualizada	65,42	64,6	32
Cenário 3 – totalmente físico	46,37	45,7	46
Cenário 4 – virtualizado em 2 máquinas diferentes	50,97	50,1	56

Tabela 4.15: Resumo resultados dos testes de iPERF.

A figura 4.53 apresenta os resultados dos testes de transmissão com a ferramenta iPERF em forma de gráfico.

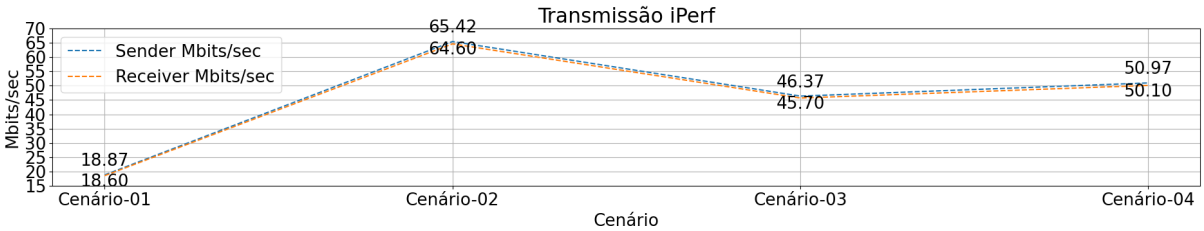


Figura 4.53: Resumo do resultado dos testes com iPERF.

Resumo dos testes de transmissão com Rsync

A tabela 4.16 traz os resultados obtidos nos testes de transmissão de dados com a ferramenta Rsync. Pelos resultados obtidos é possível notar que, assim como no teste de transmissão anterior, o cenário 1 foi onde ocorreram os piores resultados, e que a diferença do cenário 3 para os cenários 2 e 4 não foi grande.

Teste transmissão Rsync	**Mínimo Mbytes/sec**	**Média Mbytes/sec**	**Máximo Mbytes/-sec**
Cenário 1 – das redes virtualizadas na mesma máquina	1,89	2,49	2,8
Cenário 2 – cenário misto, uma rede física e uma rede virtualizada	5,67	5,77	5,99
Cenário 3 – totalmente físico	4,88	4,88	4,88
Cenário 4 – virtualizado em 2 máquinas diferentes	4,28	4,76	5,12

Tabela 4.16: Resumo resultados dos testes de Rsync.

A figura 4.54 apresenta os resultados dos testes de transmissão com a ferramenta Rsync em forma de gráfico.

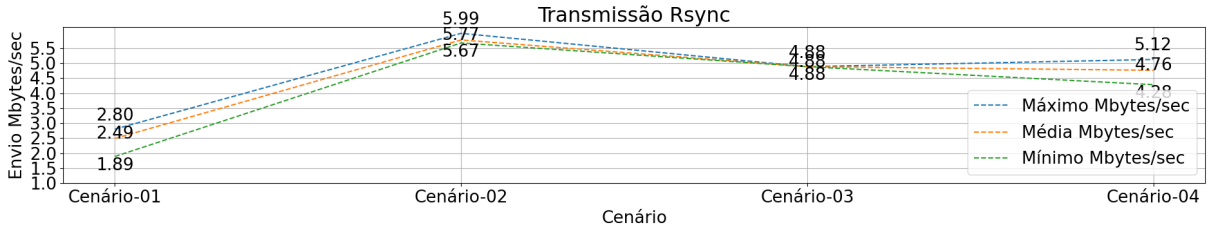


Figura 4.54: Resumo do resultado dos testes com iPERF.

Testes conexão e acesso remoto

Os testes de conexão entre as máquinas nas redes remotas demonstraram não só a possibilidade da conexão como também a estabilidade, visto que no caso do Zabbix foi possível monitorar as máquinas ativas e até mesmo gerar um alerta quando a máquina foi desligada e novo alerta quando esta foi religada, como demonstrado anteriormente na figura 4.44.

Sobre o acesso remoto, como explicado anteriormente, em máquinas Linux a ferramenta rsync utiliza o protocolo de acesso remoto SSH para realizar a transferência de arquivos, logo, como os testes de transferência funcionaram, o acesso remoto também funcionou. Sobre as máquinas Windows, foi utilizada a ferramenta Remmina para realizar o acesso remoto ao ambiente gráfico do Windows sem problemas.

4.3.2 Testes de Segurança

Os testes de segurança, como varredura de rede e os ataques de negação de serviços, não foram aplicados em todos os cenários, por tanto não é possível uma comparação. Isso ocorreu, pois os cenários iniciais tinham o objetivo de validar diferentes configurações de alocação de máquinas virtuais e físicas, ficando o último cenário, o de número 4, como alvo dos testes de segurança. Os testes de segurança realizados no cenário 4 demonstram ser possível monitorar, detectar e gerar alertas, sobre as tentativas de ataques realizadas na rede monitorada remotamente, mesmo via túnel.

5 CONCLUSÃO

Este trabalho propôs o estudo e a implementação de um esquema de interconexão para monitoramento e gerenciamento remoto de redes sobre IPv4. A escolha pela utilização do IPv4 como protocolo de endereçamento nesta proposta, se deu pelo fato de que até a presente data, este ainda é o protocolo de endereçamento mais difundido, enquanto o IPv6 vem ganhando espaço, mas não foi adotado por uma parte dos serviços encontrados na *Internet*. Como citado anteriormente, a escassez de endereços IPv4, tem obrigado os ISPs a se utilizarem de mecanismos de tradução de endereços privados para públicos de modo mais agressivo, conectando cada vez mais clientes com um único endereço público, o que tem gerado problemas de conexão em serviços que necessitam de conexões ponto-a-ponto mais estáveis ou com portas mais específicas e que não estejam entre as portas registradas.

Algumas das possíveis soluções para interconexão de redes remotas, passam pela utilização de VPNs, porém neste caso seria necessário um endereço IP público ou um domínio registrado com DNS dinâmico, recursos esses que poderiam gerar algum custo. Para obter os mesmos benefícios de uma VPN, mas sem necessidade de tais recursos, neste trabalho foi utilizado o serviço ZeroTier, pelo qual foi possível gerar uma conexão VPN *site-to-site* entre as redes remotas. Durante os testes realizados, a conexão se mostrou estável o suficiente para permitir tanto o monitoramento quanto acesso remoto e transferência de dados entre as redes. Como pode ser visto na sessão de análise dos resultados, as taxas de transferências obtidas nos testes foram um pouco prejudicadas por limitações do *hardware* utilizado, porém, ainda assim ficando dentro do esperado para redes com taxa de transferência nominal máxima de 100Mbits/segundo.

Com a aplicação de monitoramento de ativos e serviços de rede Zabbix instalada, foi possível monitorar não só os ativos presentes na rede remota, como também serviços configurados em servidor remoto, e até mesmo o estado do enlace de ligação entre as duas redes, além de disparar alertas à equipe de TI via aplicativo de mensagens.

Após a instalação de configuração da plataforma de segurança SIEM Wazuh, foi possível através do túnel, monitorar aspectos de segurança dos ativos, como vulnerabilidades do SO que devem ser corrigidas, detectar tentativas de varreduras de portas e ataques de negação de serviço. Também foi possível configurar o disparo de alertas a membros de grupos em aplicativos de mensagens.

Conclui-se, portanto, que foi possível implementar uma solução para a interconexão das redes sobre IPv4 sem custos adicionais. Através da conexão criada, foi possível, com um conjunto de soluções em *software* gratuito, monitorar questões operacionais e de segurança dos ativos presentes na rede remota, também foi possível, como demonstrado nos testes, acessar as máquinas remotamente para casos de necessidade de manutenção e a transferência arquivos entre as redes, assim como alertar administradores em caso de problemas.

Todos os recursos testados e apresentados neste trabalho, indicam a viabilidade da criação de centrais de monitoramento remoto, capazes de monitorar várias redes simultaneamente, e tornam possível a oferta do monitoramento e gerenciamento de redes como um serviço a ser prestado.

5.1 TRABALHOS FUTUROS

Como sugestão para trabalhos futuros, é possível considerar a implementação da conexão entre as redes remotas utilizando o protocolo de endereçamento IPv6, o que pode possibilitar a implementação de uma VPN direta, sem a necessidade de serviços de terceiros. Sendo que tal abordagem, levaria a uma discussão mais profunda sobre as consequências em relação a aspectos de segurança, em se ter uma rede com dispositivos realmente conectados diretamente à *Internet* e como mitigar os riscos.

REFERÊNCIAS BIBLIOGRÁFICAS

ABDI/FGV. *Maturidade Digital das MPEs Brasileiras*. 2021. Disponível em: <https://api.abdi.com.br/file-manager/upload/files/Mapa_da_Digitaliza%C3%A7%C3%A3o_das_MPEs_Brasileiras__1___1_.pdf>.

AKAMAI-TECHNOLOGIES. *O que é um ataque DDoS Slowloris?* 2023. Acessado em: 20/07/2023. Disponível em: <https://www.akamai.com/pt/glossary/what-is-a-slowloris-ddos-attack>.

ANATEL. *Grupo de Trabalho para implantação do protocolo IPVersão 6 nas redes das Prestadoras de Serviços de Telecomunicações Relatório Final de Atividades*. 2014. Acessado em: 29/04/2023. Disponível em: <https://www.anatel.gov.br/Portal/verificaDocumentos/documento.asp?numeroPublicacao=325769>.

ASF, A. S. F. *Apache HTTP Server Project*. 2023. Acessado em: 21/04/2023. Disponível em: <https://httpd.apache.org/>.

BASTA, A.; BASTA, N.; BROWN, M.; ALMEIDA, L. de; GONÇALVES, R. de L. *Segurança de Computadores E Teste de Invasão*. Cengage Learning, 2014. ISBN 9788522121366. Disponível em: <https://books.google.com.br/books?id=eVihzgEACAAJ>.

CALDAS, E. de O.; OLIVEIRA, D. M. de; MARQUES, G. dos S.; DEUS, F. E. G. de; MENDONCA, F. L.; SOUSA., R. T. D. Aplicativo para avaliação de condução segura de usuários de veículos automotores por meio de inteligência artificial para benefícios em seguros veiculares. *Conferências Ibero-Americanas WWW/Internet e Computação Aplicada 2023, Vilamoura. Lisboa: IADIS Press*, 2023.

COMER, D. *Redes de Computadores e Internet - 6.ed.* [S.l.]: Bookman Editora, 2016. <https://integrada.minhabiblioteca.com.br/#/books/9788582603734/>(Acessado: 29/04/2023). ISBN 9788582603734.

DOCKER. *Use containers to Build, Share and Run your applications*. 2023. Acessado em: 21/04/2023. Disponível em: <https://www.docker.com/resources/what-container/>.

EMBEDD. *DD-WRT About*. 2023. Acessado em: 28/05/2023. Disponível em: <https://dd-wrt.com/about/>.

FENCING, E. S. *pfSense Overview*. 2023. Acessado em: 10/05/2023. Disponível em: <https://www.pfsense.org/about-pfsense/>.

GNS3. *Getting Started with GNS3*. 2023. Acessado em: 21/04/2023. Disponível em: <https://docs.gns3.com/docs/>.

GOSENHEIMER, A. C. C.; NOGUEIRA, F. L. G. *Avaliação e teste de ataques cibernéticos via ferramenta de EDR*. [S.l.]: Universidade de Brasília, 2022.

HOLANDA, L. P. de; SILVA, P. H. N. da. *Proposta prática de laboratório virtual para o gerenciamento de métricas de desempenho, eventos e informações de segurança em ambiente de alta disponibilidade*. [S.l.]: Universidade de Brasília, 2022.

INSECURE.ORG. *nmap.org*. 2023. Acessado em: 19/07/2023. Disponível em: <https://nmap.org/>.

IPERF.FR. *SIMET*. 2023. Acessado em: 13/06/2023. Disponível em: <https://beta.simet.nic.br/>.

KASPERSKY. *What Is Endpoint Detection and Response?* 2023. Acessado em: 25/05/2023. Disponível em: <https://www.kaspersky.com.br/resource-center/preemptive-safety/endpoint-detection-and-response>.

KLUSAITė, L. *Os Tipos dos Melhores Protocolos VPN*. 2020. Acessado em: 21/04/2023. Disponível em: <https://nordvpn.com/pt-br/blog/protocolos-vpn/>.

NIC-BR. *SIMET*. 2023. Acessado em: 13/06/2023. Disponível em: <https://beta.simet.nic.br/>.

NIC.BR. *O que é CGNAT e como isso pode afetar sua conexão de internet*. 2018. Acessado em: 29/04/2023. Disponível em: <https://nic.br/noticia/na-midia/o-que-e-cgnat-e-como-isso-pode-afetar-sua-conexao-de-internet/>.

NIC.BR. *IPv6 move 45 porcento da Internet no Brasil, mas faltam roteadores e conteúdos*. 2023. Acessado em: 28/04/2023. Disponível em: <https://www.nic.br/noticia/na-midia/ipv6-move-45-da-internet-no-brasil-mas-faltam-roteadores-e-conteudos/>.

OLIVEIRA, D. M. de; FILHO, F. D. C.; MENDONCA, F. L.; NZE, G. D. A.; SILVA, D. A.; SOUSA., R. T. D. Arquitetura para monitoramento e gerenciamento remoto de redes como prestação de serviços. *Conferências Ibero-Americanas WWW/Internet e Computação Aplicada 2022, Vilamoura. Lisboa: IADIS Press*, 2022.

PEERSPOT. *Best Firewall Software*. 2023. Acessado em: 10/05/2023. Disponível em: <https://www.peerspot.com/categories/firewalls>.

QEMU. *Getting Started with GNS3*. 2022. Acessado em: 21/04/2023. Disponível em: <https://www.qemu.org/docs/master/about/index.html>.

RFC1918. *Address Allocation for Private Internets*. [S.l.], 1996. <http://www.rfc-editor.org/rfc/rfc1918.txt>. Disponível em: <http://www.rfc-editor.org/rfc/rfc1918.txt>.

RFC2663. *IP Network Address Translator (NAT) Terminology and Considerations*. [S.l.], 1999. <http://www.rfc-editor.org/rfc/rfc2663.txt>. Disponível em: <http://www.rfc-editor.org/rfc/rfc2663.txt>.

RFC6598. *IANA-Reserved IPv4 Prefix for Shared Address Space*. [S.l.], 2012. <http://www.rfc-editor.org/rfc/rfc6598.txt>. Disponível em: <http://www.rfc-editor.org/rfc/rfc6598.txt>.

RSYNC-SAMBA.ORG. *DESCRIPTION*. 2023. Acessado em: 13/06/2023. Disponível em: <https://download.samba.org/pub/rsync/rsync.1>.

SPALTER, M. *What is Carrier Grade NAT (CGNAT) ?* 2022. Acessado em: 20/04/2023. Disponível em: <https://www.draytek.co.uk/information/blog/what-is-cgnat/>.

SáNCHEZ, J. J. *Integrating Telegram with Wazuh*. 2020. Acessado em: 8/06/2023. Disponível em: <https://medium.com/@jesusjimsa_12801/integrating-telegram-with-wazuh-4d8db91025f>.

W3TECHS. *Historical trends in the usage statistics of web servers*. 2023. Acessado em: 21/04/2023. Disponível em: <https://w3techs.com/technologies/history_overview/web_server>.

WAZUH-INC. *Getting started with Wazuh*. 2023. Acessado em: 25/05/2023. Disponível em: <https://documentation.wazuh.com/current/getting-started/index.html>.

WIRESHARK-FOUNDATION. *about*. 2023. Acessado em: 25/05/2023. Disponível em: <https://www.wireshark.org/about.html>.

YALTIRAKLI, G. Slowloris. *github.com*, 2015. Acessado em: 20/07/2023. Disponível em: <https://github.com/gkbrk/slowloris>.

ZABBIX-SIA. *What is Zabbix*. 2023. Acessado em: 25/05/2023. Disponível em: <https://www.zabbix.com/documentation/current/en/manual/introduction/about>.

ZEROTIER-INC. *Protocol Design Whitepaper*. 2023. Acessado em: 25/05/2023. Disponível em: <https://docs.zerotier.com/zerotier/manual>.

6 APÊNDICES

Neste capítulo apresentamos algumas configurações mais específicas e requerem maiores explicações.

6.1 CONFIGURAÇÕES

6.1.1 DD-WRT Virtualização

Como dito anteriormente o DD-WRT é um *firmware* para roteadores, para virtualizar o sistema em uma máquina *descktop* foi necessário utilizar a versão compatível com processadores x86, que atualmente não tem mais suporte do projeto. Seguem os passos para instalação do sistema em uma máquina virtual utilizando Qemu.

Baixar a imagem de disco do sistema disponível no link: https://dd-wrt.com/support/other-downloads/?path=obsolete%2Fbeta%2FX86%2F08-06-2007%2Ffree%2F

Converter a imagem de disco para o formato do qcow2 utilizado pelo Qemu.

```
qemu-img convert dd-wrt_public_vga.image dd-wrt_public_vga_1.qcow2
```

A imagem vem no tamanho mínimo e precisa ser expandida para funcionar

```
qemu-img resize  dd-wrt_public_vga_1.qcow2 200M
```

Criar a máquina no emulador GNS3 via interface gráfica, e adicionar a imagem de disco DD-WRT, seguindo os passos.

```
GNS3 > Edit > Preferences > Qemu VMs > New >
adicionar um nome > next >
configurar quantidade de memoria > next >
console type > VNC >
New Image > Browse > dd-wrt_public_vga_1.qcow2 > Finish
```

6.1.2 pFsense Zerotier

Como o pFsense é uma versão modificada do FreeBSD, não foi possível compilar os instaladores na própria máquina. Foi preciso instalar uma versão completa do FreeBSD em uma máquina virtual, para compilar os instaladores, depois transferir os instaladores para as máquinas pFsense do cenário e fazer as instalações.

Esta sub-sessão descreve o passo a passo para preparar o ambiente e realizar a compilação dos instaladores.

Preparando o ambiente:

A figura 6.1 mostra saida do comando CLI para verificar a versão do FreeBSD utilizada no pFsense.

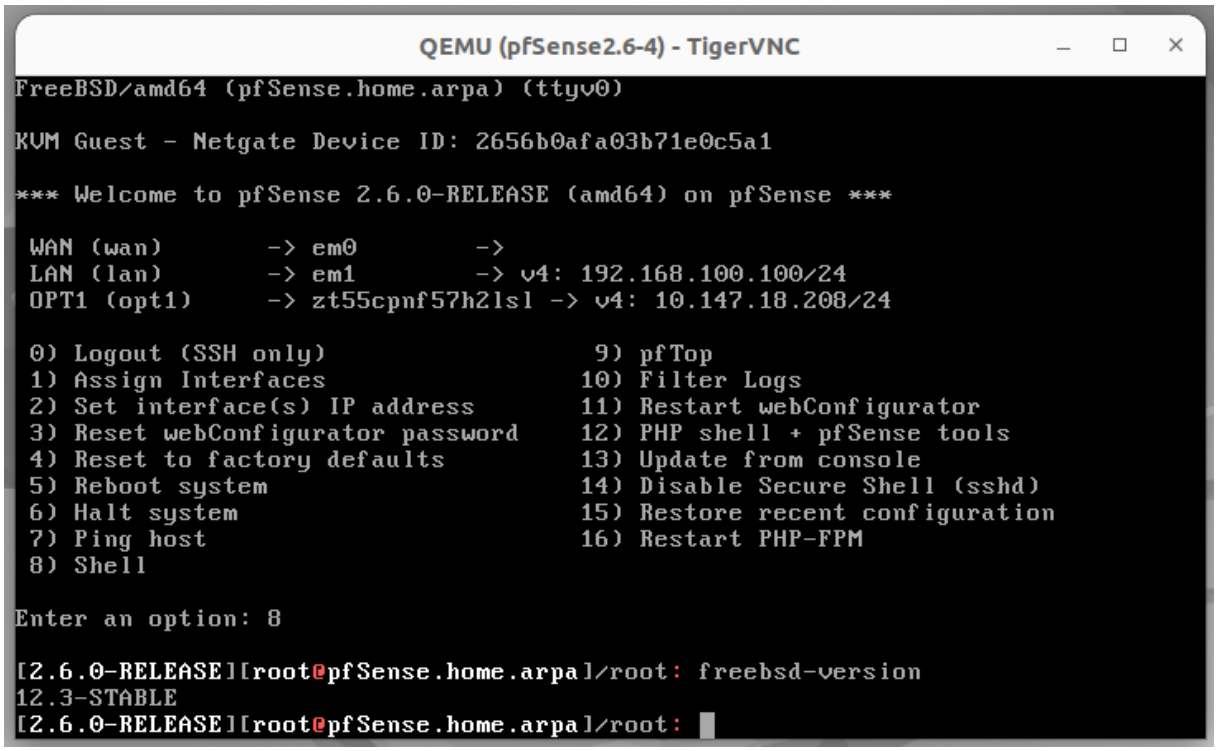


Figura 6.1: Versão do FeeBSD utilizada no pFsense

```
#Checar a versao do FreeBSD no pFsense
freebsd-version
12.3-STABLE

#Baixar a imagem de instalacao do sistema em:
#https://download.freebsd.org/releases/i386/i386/ISO-IMAGES/12.4/

#Instalar o sistema operacional em uma maquina virtual
#Acessar o sistema
```

```
#Instalando o ports
portsnap fetch extract
portsnap fetch update
```

Criar o instalador do gerenciador de conexoes do ZeroTier, zerotier-cli:

```
#Entrar na pasta do ports zerotier

cd /usr/ports/net/zerotier/

#Compilar
make clean ; make package

#Entrar na pasta work
cd work/pkg

#Verificar que o instalador foi criado

ls -lh

Instalador zerotier-1.10.6.pkg criado
```

A figura 6.2 mostra o instalador zerotier-1.10.6.pkg foi criado

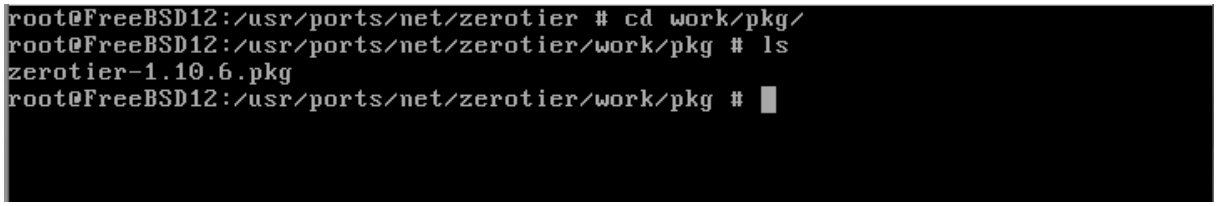


Figura 6.2: Arquivo de instalação zerotier-cli

Criar o instalador da interface grafica do ZeroTier:

```
#Criar o instalador do zerotier GUI que vai gerar a interface grafica
#Entrar na pasta home
cd ~

#Instalar o git
pkg install git

#Baixar os aquivos do git
git clone https://github.com/pfsense/FreeBSD-ports.git
```

```
#Criar o arquivo "/etc/make.conf" caso nao exista e adicionar a linha "
    ALLOW_UNSUPPORTED_SYSTEM=YES"
echo "ALLOW_UNSUPPORTED_SYSTEM=YES" > /etc/make

#Criar uma pasta para clonar o repositorio
mkdir pfsense-pkg-zerotier
cd pfsense-pkg-zerotier

#Clonar o repositorio
git clone https://github.com/ChanceM/pfSense-pkg-zerotier.git

#Entrar na pasta baixada
cd pfsense-pkg-zerotier

#Compilar
make clean ; make package

#Entar na pasra work/pkg
cd work/pkg

#Verificar que o instalador foi criado

ls

Instalador pfsense-pkg-zerotier-0.001.pkg criado
```

A figura 6.3 mostra o instalador pfsense-pkg-zerotier-0.001.pkg foi criado

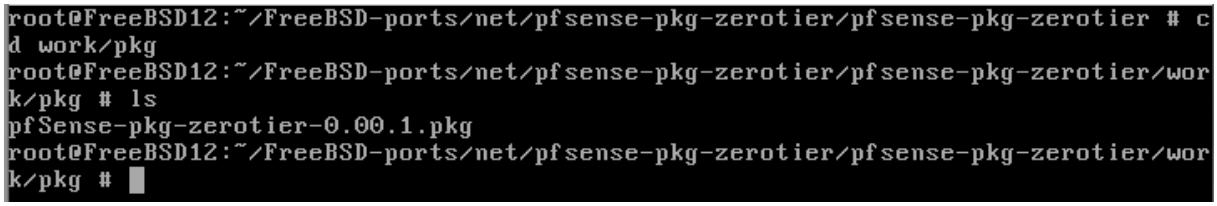


Figura 6.3: Arquivo de instalação zerotier-gui

Com os dois instaladores criados, transferir os dois para a máquina pFsense e fazer a instalação.

```
pkg install zerotier-1.10.6.pkg
pkg install pfsense-pkg-zerotier-0.001.pkg
```

6.1.3 pFsense Wazuh

O pFsense não possui pacote de instalação do agente wazuh em seu repositório oficial, então utilizamos o pacote do FreeBSD, que assim como os anteriores precisa ser compilado em uma máquina FreeBSD completa, e depois transferido para o pFsense para a instalação.

```
#Criar o instalador do Wazuh agent
#Utilizando mesma maquina FreeBSD utilizada anteriormente

#Entrar na pasta ports Wazuh
cd /usr/var/sercurity/wazuh-agent

#Compilar
make clean ; make package

#Entar na pasra work/pkg
cd work/pkg

#Verificar que o instalador foi criado
ls

#Instalador wazuh-agent-4.4.3.pkg criado

#Transferir o instalador para a maquina pFsense e fazer a instalacao.
pkg install wazuh-agent-4.4.3.pkg
```

A figura 6.4 mostra o instalador wazuh-agent-4.4.3.pkg foi criado

```
root@FreeBSD12:/usr/ports/security/wazuh-agent # cd work/pkg
root@FreeBSD12:/usr/ports/security/wazuh-agent/work/pkg # ls
wazuh-agent-4.4.3.pkg
root@FreeBSD12:/usr/ports/security/wazuh-agent/work/pkg #
```

Figura 6.4: Arquivo de instalação wazuh-agent

6.1.4 Wazuh Telegram

Para integração do Wazuh com o Telegram, é necessário utilizar um *scritp* para enviar os alertas via API do Telegram. O *script* utilizado foi aptado de (SáNCHEZ, 2020), e modificado para enviar mais campos com informações e para enviar apenas alertas com nível acima de 4. Seguem os passos para a instação:

```
#Instalar o modulo requests
pip3 install requests
```

```
#Criar o script na pasta de integracao do wazuh
nano /var/ossec/integrations/custum-telegram

#Acertar as permicoes e usuarios
chmod 750 /var/ossec/integrations/custum-telegram
chown root:wazuh /var/ossec/integrations/custum-telegram

#Reiniciar o wazuh manager
systemctl restart wazu-manager
```

Segue *scritp*:

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import sys
import json

try:
    import requests
except Exception:
    print("No module 'requests' found. Install: pip3 install requests")
    sys.exit(1)

CHAT_ID = "Insert Chat ID here"


def create_message(alert_json):
    # Get alert information
    title = alert_json['rule']['description'] if 'description' in alert_json['rule'] else ''
    description = alert_json['full_log'] if 'full_log' in alert_json else ''
    description.replace("\\n", "\n")
    alert_level = alert_json['rule']['level'] if 'level' in alert_json['rule'] else ''
    groups = ', '.join(alert_json['rule']['groups']) if 'groups' in alert_json['rule'] else ''
    rule_id = alert_json['rule']['id'] if 'rule' in alert_json else ''
    agent_name = alert_json['agent']['name'] if 'name' in alert_json['agent'] else ''
    agent_id = alert_json['agent']['id'] if 'id' in alert_json['agent'] else ''
    agent_ip = alert_json['agent']['ip'] if 'ip' in alert_json['agent'] else ''


    # Format message with markdown
    msg_content = f'*Wazuh Server Alert*\n\n'
```

```
        msg_content += f'*{title}*\n\n'
        msg_content += f'_{description}_\n'
        msg_content += f'*Groups:* {groups}\n' if len(groups) > 0 else ''
        msg_content += f'*Rule:* {rule_id} (Level {alert_level})\n'
        msg_content += f'*Agent Name:* {agent_name} ({agent_id})\n' if len(agent_name) > 0 else ''
        msg_content += f'*Agent IP:* {agent_ip}\n' if len(agent_ip) > 0 else ''

        msg_data = {}
        msg_data['chat_id'] = CHAT_ID
        msg_data['text'] = msg_content
        msg_data['parse_mode'] = 'markdown'

        # Debug information
        with open('/var/ossec/logs/integrations.log', 'a') as f:
            f.write(f'MSG: {msg_data}\n')

        return json.dumps(msg_data)


    # Read configuration parameters
    alert_file = open(sys.argv[1])
    hook_url = sys.argv[3]

    # Read the alert file
    alert_json = json.loads(alert_file.read())
    alert_file.close()

    if alert_json['rule']['level'] > 5:

      # Send the request
      msg_data = create_message(alert_json)
      headers = {'content-type': 'application/json', 'Accept-Charset': 'UTF-8'}
      response = requests.post(hook_url, headers=headers, data=msg_data)

    # Debug information
    with open('/var/ossec/logs/integrations.log', 'a') as f:
        f.write(f'RESPONSE: {response}\n')

    sys.exit(0)
```

6.2 TESTES

6.2.1 Cenário 2

Script de testes do cenário 2:

```bash
#!/bin/bash

#arquivo de log
log="/home/aluno/logIperf_tcp.txt"              #arquivo de log

echo -e "=========================\n" >> $log   #separador para faciliar a leitura do log
echo "Time " $(date) >> $log                     #adicionar a hora do teste
iperf3 -c 192.168.100.10 >> $log                 #teste de envio e recebimento em TCP via rede zerotier
echo iPerf3 finalizado


echo ping iniciado
log='/home/aluno/log_ping.txt'
echo -e "=========================\n" >> $log
echo "Time " $(date) >> $log
ping 192.168.100.10 -c 100 >> $log
echo ping finalizado
```

6.2.2 Cenário 4

Saída da varredura com nmap

lognmap1.txt

```
Starting Nmap 7.93  https://nmap.org  at 2023-07-19 14:08 EDT
Nmap scan report for 192.168.100.1
Host is up 0.00065s latency.
Not shown: 997 closed tcp ports reset
PORT   STATE SERVICE VERSION
22/tcp open  ssh     Dropbear sshd 2020.81 protocol 2.0
23/tcp open  telnet
80/tcp open  http    httpd
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT     ADDRESS
1   0.65 ms 192.168.100.1

Nmap scan report for 192.168.100.10
Host is up 0.00044s latency.
Not shown: 998 closed tcp ports reset
PORT     STATE SERVICE       VERSION
22/tcp   open  ssh           OpenSSH 7.6p1 Ubuntu 4ubuntu0.7 Ubuntu Linux; protocol 2.0
3389/tcp open  ms-wbt-server xrdp
Running: Linux 4.X.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.6
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT     ADDRESS
1   0.44 ms 192.168.100.10
```

```
Nmap scan report for 192.168.100.20
Host is up 0.00044s latency.
Not shown: 990 closed tcp ports reset
PORT      STATE SERVICE      VERSION
135/tcp   open  msrpc        Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds Windows 7 Professional 7601
Service Pack 1 microsoft-ds workgroup: WORKGROUP
3389/tcp  open  tcpwrapped
49152/tcp open  msrpc        Microsoft Windows RPC
49153/tcp open  msrpc        Microsoft Windows RPC
49154/tcp open  msrpc        Microsoft Windows RPC
49156/tcp open  msrpc        Microsoft Windows RPC
49157/tcp open  msrpc        Microsoft Windows RPC
49160/tcp open  msrpc        Microsoft Windows RPC
MAC Address: 0C:3E:98:D4:00:00 Unknown
Running: Microsoft Windows 7008.1
OS CPE: cpe:/o:microsoft:windows_7::- cpe:/o:microsoft:windows_7::sp1

cpe:/o:microsoft:windows_server_2008::sp1
cpe:/o:microsoft:windows_server_2008:r2
cpe:/o:microsoft:windows_8 cpe:/o:microsoft:windows_8.1
OS details: Microsoft Windows 7 SP0 - SP1,
Windows Server 2008 SP1, Windows Server 2008 R2,
Windows 8, or Windows 8.1 Update 1
Network Distance: 1 hop
Service Info: Host: W7; OS: Windows; CPE: cpe:/o:microsoft:windows

TRACEROUTE
HOP RTT     ADDRESS
1   0.44 ms 192.168.100.20

Nmap scan report for 192.168.100.30
Host is up 0.00026s latency.
All 1000 scanned ports on 192.168.100.30 are in ignored states.
Not shown: 1000 closed tcp ports reset
MAC Address: 2A:D5:9A:33:24:69 Unknown
Too many fingerprints match this host to give specific OS details
Network Distance: 1 hop

TRACEROUTE
HOP RTT     ADDRESS
1   0.25 ms 192.168.100.30

Nmap scan report for 192.168.100.100
Host is up 0.00059s latency.
Not shown: 996 filtered tcp ports no-response
PORT    STATE SERVICE  VERSION
22/tcp  open  ssh      OpenSSH 7.9 protocol 2.0
53/tcp  open  domain   generic dns response: REFUSED
80/tcp  open  http     nginx
```

```
443/tcp open  ssl/http nginx
Aggressive OS guesses: FreeBSD 11.2-RELEASE 93%
No exact OS matches for host test conditions non-ideal.
Network Distance: 1 hop

TRACEROUTE
HOP RTT     ADDRESS
1   0.59 ms 192.168.100.100

Nmap scan report for 192.168.100.200
Host is up 0.00057s latency.
Not shown: 998 closed tcp ports reset
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.5 Ubuntu Linux; protocol 2.0
80/tcp open  http    Apache httpd 2.4.41 Ubuntu
MAC Address: 0C:F1:63:2C:00:00 Unknown
Running: Linux 4.X.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.6
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT     ADDRESS
1   0.57 ms 192.168.100.200

OS and Service detection performed. Please report any incorrect
results at https://nmap.org/submit/ .
Nmap done: 256 IP addresses 7 hosts up scanned in 242.02 seconds
```