WILLIAM DOS SANTOS ABREU

ADMINISTRAÇÃO DE SERVIÇOS DE REDE EM UM PROVEDOR DE ACESSO À INTERNET

LAVRAS – MG

2020

WILLIAM DOS SANTOS ABREU

ADMINISTRAÇÃO DE SERVIÇOS DE REDE EM UM PROVEDOR DE ACESSO À INTERNET

Relatório de estágio supervisionado apresentado à Universidade Federal de Lavras, como parte das exigências do Curso de Ciência da Computação, para a obtenção do título de Bacharel.

Prof. Dr. Neumar Costa Malheiros
Orientador

LAVRAS – MG

2020

WILLIAM DOS SANTOS ABREU

ADMINISTRAÇÃO DE SERVIÇOS DE REDE EM UM PROVEDOR DE ACESSO À INTERNET

Relatório de estágio supervisionado apresentado à Universidade Federal de Lavras, como parte das exigências do Curso de Ciência da Computação, para a obtenção do título de Bacharel.

APROVADA em 4 de setembro de 2020.

Prof. Dr. Neumar Costa Malheiros	UFLA
Prof. Dr. Luiz Henrique Andrade Correia	UFLA
Prof. Dr. Hermes Pimenta de Moraes Júnior	UFLA

Prof. Dr. Neumar Costa Malheiros
Orientador

LAVRAS – MG
2020

Dedico aos meus pais, Vera e Vicente.

AGRADECIMENTOS

Agradeço à minha família, em especial aos meus pais, Vera e Vicente, e ao meu irmão, Wesley, pelo apoio pessoal nessa etapa.

Agradeço aos diretores da Minasnet, André e Admilson, por terem oferecido a oportunidade de estagiar na empresa.

Agradeço ao meu orientador, Neumar, pelo auxílio e pela paciência no desenvolvimento deste trabalho.

Agradeço ao meu gerente, Fábio, pelo conhecimento técnico ensinado e pelo apoio na execução das tarefas.

Agradeço aos colegas do NOC da empresa, pela parceria e pela troca de experiências no ambiente de trabalho.

A informática e as telecomunicações serão para o século XXI
o que as rodovias foram para o século XX. (Bill Clinton)

RESUMO

Os provedores de acesso à internet têm o desafio de manter ininterrupto o serviço de rede aos assinantes, garantindo velocidade e estabilidade na conexão. Embora esse seja o objetivo comercial dessas empresas, existem vários problemas operacionais que demandam a atenção dos administradores da rede para garantir pleno funcionamento do provedor. Um dos problemas enfrentados é a escassez de endereços IPv4, que impõe limitações na conectividade à internet. Outro problema tange à segurança da rede, uma vez que dispositivos conectados estão suscetíveis a ataques, que podem comprometer a disponibilidade do serviço. O objetivo deste trabalho foi desenvolver soluções para problemas que envolvem a conectividade dos assinantes à internet. Essas soluções obedecem a lei do Marco Civil da Internet e mantêm os acordos cooperativos firmados entre os sistemas autônomos e entre o CERT.br, para manutenção da segurança da Internet brasileira. As atividades foram desenvolvidas em estágio supervisionado no centro de operações de rede da Minasnet Telecomunicações LTDA, operadora de internet que presta serviço em 21 cidades no sul de Minas Gerais. Neste estágio foi feito o dimensionamento de redes IPv4 e foi implementado CGNAT como solução para a escassez de endereços (enquanto o IPv6 não é uma tecnologia amplamente utilizada na internet como um todo). Além disso, foram aplicados recursos de segurança à rede, sendo utilizados dois mecanismos: VPN, para controlar o acesso à intranet do provedor, e firewall, para bloquear conexões suspeitas com a finalidade de mitigar ataques de negação de serviço. Este trabalho possibilitou o aprofundamento prático na área de Sistemas de Computação, especificamente nos assuntos de Redes, Sistemas Distribuídos e Segurança Computacional, servindo de ponte entre a academia e o conhecimento técnico-científico utilizado no mercado.

Palavras-chave: CGNAT. VPN SSL. Controle de acesso. Regras de firewall.

ABSTRACT

The internet service providers get the challenge in keeping uninterrupted the subscribers' networking service, ensuring speed and reliability in the internet connection. Although this is the comercial goal for these enterprises, there are many operational problems that need the network administrators attention to guarantee plain working of the internet provider. One of the problems faced is the IPv4 addresses exhaustion, what imposes limitations on network connectivity. Another pertain to network security, since the connected devices are susceptible to attacks, that can compromise de service availability. The purpose of this work in particular was develop solutions for problems that involve the subscribers internet connectivity, obeying the Civil Rights Framework for the Internet and keeping the cooperative agreements established between the autonomous systems and the CERT.br for supporting the brazilian internet safety. The activities were developed as part of internship at Minasnet Telecomunicações LTDA's network operation center, enterprise that provides network service to 21 cities in the south of the state of Minas Gerais, where was made the sizing of IPv4 networks and was implemented CGNAT as solution for the addresses exhaustion (while IPv6 isn't a widely used technology). Besides, were applied security resources in the network, using VPN for controlling the access to the provider's intranet and firewall for blocking suspects connections for the purpose of mitigating denial of service attacks. The work made possible the practice deepen in the Computer Systems area, closely at the subjects of Networking, Distributed Systems and Computer Security, building a bridge between the academia and the market technological knowledge.

Keywords: CGNAT. SSL VPN. Access control. Firewall rules.

LISTA DE FIGURAS

Figura 2.1 – Mapa global de Regional Internet Registry (RIR) da IANA. 18
Figura 2.2 – Ilustração do modelo de CGNAT horizontal. 21
Figura 2.3 – Ilustração do modelo de CGNAT vertical. 21
Figura 2.4 – Modelo de topologia de um ISP. 25
Figura 3.1 – Template do comando de configuração de netmap no RouterOS. 32
Figura 3.2 – Árvore binária das subdivisões recursivas de um /22 até um /25. 33
Figura 3.3 – Exemplo de uso do programa py-cgnat para geração de CGNAT para RouterOS. 33
Figura 3.4 – Exemplo de uso do programa py-cgnat para traduções de IP privado para público. 34
Figura 3.5 – Exemplo de uso do programa py-cgnat para tradução de IP público para privado. 34
Figura 4.1 – Exemplo de consulta à DNSBL CBL via terminal Linux. 37
Figura 4.2 – Arquitetura de um honeypot para detecção de spam. 38
Figura 4.3 – Exemplo de uso do nmap para varredura completa de portas. 39
Figura 4.4 – Exemplo de criação de address-list para o firewall do RouterOS. 45
Figura 4.5 – Regra de firewall para controle de acesso aos CPEs. 46
Figura 4.6 – Regra de firewall para bloqueio de acesso à rede privada do provedor. . . . 46
Figura 4.7 – Regra de firewall para correção da vulnerabilidade por SOCKS notificada pelo CERT.br. 47

LISTA DE GRÁFICOS

Gráfico 4.1 – Contagem de IPs do AS listados na CBL entre abril e setembro de 2019. . 38

LISTA DE QUADROS

Quadro 3.1 – Exemplo de planilha para controle de CGNAT. 29
Quadro 3.2 – Mapeamento direto entre sub-redes internas/externas usando CGNAT 1:32. 31

LISTA DE SIGLAS

ABR	Area Border Router
AS	Autonomous System
B-RAS	Broadband Remote Access Server
CBL	Composite Blocking List
CERT	Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança
CGNAT	Carrier-grade NAT
CIDR	Classless Inter-Domain Routing
CLI	Command line interface
CPE	Customer Premises Equipment
DDoS	Distributed Denial of Service
DNS	Domain Name System
DNSBL	DNS Blacklist
IP	Internet Protocol
IPv4	Internet Protocol version 4
ISP	Internet Service Provider
NAT	Network Address Translation
NOC	Network Operations Center
OSI	Open System Interconnection
OSPF	Open Shortest Path First
PPPoE	Point-to-Point Protocol over Ethernet
RADIUS	Remote Authentication Dial In User Service
RBL	Real-time Blackhole List
RFC	Request for Comment
SSL	Secure Socket Layer
TCP	Transport Control Protocol
TI	Tecnologia da Informação
TIC	Tecnologia da Informação e Comunicação
VLSM	Variable-Length Subnet Masking
VPN	Virtual Private Network
VPS	Virtual Private Server

SUMÁRIO

1 INTRODUÇÃO 13
1.1 Objetivos do estágio 13
1.2 Caracterização do ambiente de trabalho 14
1.3 Capacitações e treinamentos 15
1.4 Estrutura do documento 15
2 REFERENCIAL TEÓRICO 17
2.1 Endereçamento IP 17
2.1.1 Mitigando a escassez do IPv4 com CGNAT 18
2.1.2 Aspectos legais quanto ao uso de NAT 19
2.1.3 Técnicas de CGNAT 20
2.1.4 Desvantagens do NAT 20
2.1.5 Migração para IPv6 22
2.2 Segurança de ambientes de rede 22
2.2.1 Firewall 23
2.2.2 VPN 23
2.3 Topologia de um ISP 24
3 ENDEREÇAMENTO IP 26
3.1 Rede IP da Minasnet 26
3.2 Metodologia para implementação de CGNAT 27
3.2.1 Implementação de CGNAT no RouterOS 30
3.2.2 Utilização do utilitário py-cgnat 33
3.3 Considerações sobre o uso de CGNAT 34
3.3.1 Dificuldades encontradas na implantação 35
4 SEGURANÇA DE REDE 36
4.1 IP Blacklist 36
4.2 Vulnerabilidades de rede 39
4.3 Implantação de VPN 40
4.3.1 Configuração do serviço OpenVPN 41
4.3.2 Considerações sobre o servidor OpenVPN 42
4.4 Implementação de firewall 42
4.4.1 Regras de firewall no RouterOS 44

4.4.2 Considerações sobre o firewall 47
5 CONCLUSÃO 48
REFERÊNCIAS 50

1 INTRODUÇÃO

A internet tornou-se um dos principais meios de comunicação utilizados pela população. Esse fato é sustentado pelos dados recentemente divulgados pela DataReportal, que mostram o número de usuários na internet brasileira crescendo a cada ano. No relatório de janeiro de 2020, o portal contabilizou 150,4 milhões de navegantes na web no país, um valor que cresceu 6% em um ano, com 8,5 milhões de usuários novos em relação a 2019 (KEPIOS, 2020). Os dados mais recentes mostram que 71% dos brasileiros têm acesso a esse meio de comunicação.

Dessa forma, manter operacional toda a infraestrutura que sustenta essa crescente demanda por acesso à internet é uma tarefa que exige profissionais habilitados. Por isso, existem os profissionais que administram os serviços de rede das operadoras, implantando e mantendo as redes de acesso de seus assinantes com a maior disponibilidade possível. Para cada aplicação disponível na web aos usuários finais, existe toda uma infraestrutura que permite que os dados trafeguem, em uma fração de segundos, através de diversos tipos de enlace (como fibra óptica), para manter o mundo inteiro conectado.

1.1 Objetivos do estágio

Este trabalho foi realizado durante estágio na empresa Minasnet Telecomunicações LTDA. O objetivo do estágio é desenvolver e implantar mecanismos para operação e manutenção de serviços de rede com ênfase em segurança e desempenho. As atividades de estágio possibilitaram ao estudante atuar na operação e na manutenção do serviço de internet, se envolvendo diretamente com o funcionamento de um provedor de internet e com o trabalho dos administradores e dos analistas de redes na organização.

A princípio, o plano de trabalho consistia em: realizar monitoramento dos ativos de rede e dos blocos de endereços IP listados em *blacklist*; realizar análise dos procedimentos realizados pelos técnicos de atendimento; e prestar suporte aos técnicos de instalação e de infraestrutura, criando relatórios, procedimentos e documentando os processos técnicos realizados. Porém, com a competência e o desempenho do estagiário na execução das atividades propostas, além do entrosamento com a equipe e com a aptidão em lidar com novas tecnologias, outras tarefas foram sendo atribuídas ao estagiário como uma forma de evolução nos procedimentos realizados.

Assim, as atividades foram desempenhadas com o intuito de apoiar as tarefas de monitoramento de incidentes na rede e contribuir na implementação de soluções para problemas

identificados na operação dos serviços de rede. Propor melhorias em processos técnicos no setor de TI da empresa e desenvolver métodos de segurança para combate a spam e outros *worms* que se propagam pela internet também fizeram parte do trabalho. Além da execução de tarefas, o estágio também serviu como fonte de pesquisa de novos equipamentos de rede e de novas tecnologias que puderam ser aplicadas nos ambientes em produção.

1.2 Caracterização do ambiente de trabalho

A Minasnet é um ISP (*Internet Service Provider* – Provedor de Serviço de Internet), sediado na cidade de Perdões, que leva internet banda larga para 21 cidades no sul de Minas Gerais até então, sendo uma empresa fundada no ano de 2006 na mesma cidade. As atividades de estágio foram realizados, majoritariamente, no NOC (*Network Operations Center* – Centro de Operações de Rede) da empresa, sendo que algumas tarefas foram realizadas em campo e outras remotamente.

O NOC da Minasnet conta com um escritório onde trabalham os técnicos internos da empresa. A equipe interna consiste em colaboradores que desempenham os seguintes cargos: Gerente, Projetista de Rede, Administrador de Redes, Analista de Redes, Analista de Sistemas e Estagiário. O estágio foi realizado no período de março de 2019 até junho de 2020, com uma carga horária de 30 horas de trabalho semanais, cumpridas em uma jornada flexível.

Para provisionar aos clientes acesso à internet, o ISP mantém toda uma infraestrutura física em operação, composta por roteadores, switches, terminadores ópticos (OLTs) e rádios digitais. Esses equipamentos são configurados, gerenciados e monitorados pela equipe do NOC, sendo utilizadas, conforme fornecido pelo fabricante, aplicações gráficas desktop ou via web para gerência e configuração dos dispositivos ou então acesso por terminal de comando através de protocolos SSH ou Telnet. O monitoramento é feito em tempo real através de plataforma de software configurada para apresentar dados na forma de um painel de controle (*dashboard*) e enviar notificações com alertas críticos para mensageiro instantâneo através de *bot*.

Exemplificando, a aplicação desktop utilizada para equipamentos MikroTik é o Winbox, como também está disponível acesso aos equipamentos da marca através de aplicação web, de terminal Telnet ou SSH. Para monitoramento é utilizado o Zabbix, com disparo de mensagens através do Telegram, além do Video Wall no NOC com um *dashboard* completo para o monitoramento em tempo real da rede, a fim de detectar ou prever problemas, utilizando-se de gráficos de consumo de banda e de alertas de enlace desconectado, por exemplo.

A comunicação oficial da empresa é feita através do mensageiro Telegram, do e-mail institucional e dos ramais VoIP (telefone IP) que cada colaborador possui. A gestão das tarefas realizadas pela equipe interna é feita através do Kanban, aplicado por meio da ferramenta Trello.

1.3 Capacitações e treinamentos

Na Minasnet, após a admissão de qualquer colaborador, seja tanto trabalhador formal quanto estagiário, são realizadas capacitações e treinamentos para integrar os novatos nos processos da empresa, de acordo com a função que cada um for assumir. Para o estagiário, nas primeiras semanas de trabalho, um dos integrantes do NOC fornece uma capacitação individual expositiva, apresentando conceitos básicos de redes e da topologia do backbone da empresa, como também de procedimentos de atendimento e de suporte a clientes. São fornecidos manuais dos equipamentos que são usados e o novato é encaminhado para um treinamento para praticar os procedimentos aprendidos. O primeiro treinamento foi montar um pequeno provedor de laboratório utilizando equipamentos MikroTik, simulando o roteamento estático e dinâmico com OSPF em roteadores RB750[1] e enlaces de rádio com Groove[2], tudo realizado no primeiro dia do estágio, para se familiarizar desde então com os principais equipamentos utilizados pelo provedor.

O último treinamento oferecido no período do estágio foi o minicurso presencial, com duração de 18h, denominado “Protocolo de Roteamento OSPF e Mikrotik de Iniciante a Intermediário”, restrito aos técnicos internos da Minasnet, para treinamento de roteamento dinâmico com OSPF no MikroTik, utilizando o simulador GNS3[3], bem como dispositivos reais.

1.4 Estrutura do documento

Este relatório está estruturado da seguinte forma. No Capítulo 2, é apresentado o referencial teórico que fundamenta o que foi desenvolvido nas principais atividades do estágio, englobando os conceitos básicos de redes TCP/IP (*Transmission Control Protocol / Internet Protocol*) e VLSM (*Variable Length Subnet Masking*) dentro da administração de serviços de redes, bem como fundamentos de segurança computacional aplicados em redes. Nos dois capítulos seguintes, são detalhadas as atividades realizadas durante o trabalho no NOC da ope-

[1] Roteador MikroTik RB750 <https://mikrotik.com/product/RB750r2>.
[2] Rádio outdoor MikroTik Groove <https://mikrotik.com/product/RBGroove52HPnr2>.
[3] GNS3 <https://www.gns3.com> é um simulador completo de redes.

radora. No Capítulo 3, são abordado o endereçamento IPv4 e a implementação de CGNAT (*Carrier-grade Network Address Translation*), e, no Capítulo 4, é descrita a implementação de uma camada de segurança com Firewall e VPN (*Virtual Private Network*). Por fim, no Capítulo 5, são apresentadas as considerações finais, assim como uma discussão da relação das atividades do estágio com o curso de graduação.

2 REFERENCIAL TEÓRICO

O modelo de referência de redes, o modelo OSI (*Open System Interconnection*), foi desenvolvido no final dos anos 1970 pela Organização Internacional para Padronização (ISO) para definir a arquitetura das redes de computadores emergentes na época (KUROSE; ROSS, 2014). A arquitetura da rede é definida em camadas, que têm por objetivo garantir a independência entre os serviços oferecidos por cada uma delas. Essa independência é denominada encapsulamento, sendo que cada uma das camadas só pode se comunicar com as respectivas camadas adjacentes. O modelo OSI é dividido em 7 camadas: camada física, camada de enlace, camada de rede, camada de transporte, camada de sessão, camada de apresentação e camada de aplicação. Cada uma das camadas também costumam ser chamadas por números, sendo que a camada física é a camada 1 e camada de aplicação é a camada 7. As outras camadas seguem a mesma ordem de numeração.

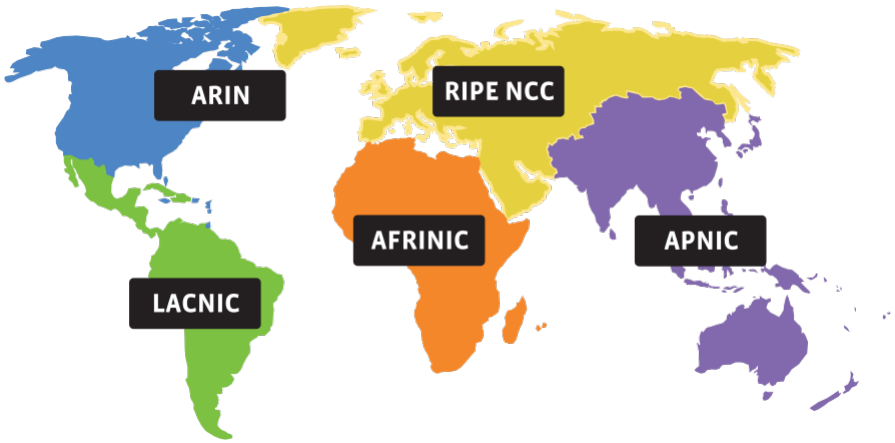
O HTTP (*Hypertext Transfer Protocol*), que é o protocolo por trás da web, é implementado no topo da pilha, na camada 7. Graças ao encapsulamento da pilha de protocolos, é possível o desenvolvimento de aplicações web sem a necessidade de preocupação com o meio de comunicação pelo qual os dados serão enviados. De acordo com o encapsulamento, uma requisição HTTP sairá do navegador da web e será encaminhada ao sistema operacional (SO). O SO estabelece a comunicação fim-a-fim entre cliente e servidor por meio de sockets TCP, que estão na camada de transporte do modelo. A partir de então, a requisição feita pela aplicação chega à placa de rede do dispositivo e será transmitida, através da internet, até o seu destino. Para isso, na camada de rede (ou camada 3) estão os esforços para que a comunicação pela internet funcione, através do *Internet Protocol* – IP.

2.1 Endereçamento IP

O IPv4 (IP versão 4) é um protocolo implementado na camada de rede, tendo como objetivo fazer o encaminhamento e o roteamento dos pacotes. Um endereço IP é um identificador único na internet e é ele que fornece identidade aos dispositivos na rede. Para que a unicidade seja garantida, é necessário um acordo centralizado para gestão dos endereços. A IANA (*Internet Assigned Numbers Authority*) é o órgão internacional responsável pela administração dos endereços, obedecendo diretrizes estabelecidas na RFC (*Request for Comments*) 2050 e delegando blocos de endereços para administração regional, divididos geograficamente nos cinco grupos apresentados na Figura 2.1. Na América do Sul, a IANA delega ao LACNIC (Registro

de Endereços da Internet para a América Latina e o Caribe) a administração dos endereços, que por sua vez delega ao NIC.br (Núcleo de Informação e Coordenação do Ponto BR) para controle regional no Brasil, sendo este último o órgão recorrido pelos ISPs nacionais para obtenção de blocos de endereços (IANA, 2020).

Figura 2.1 – Mapa global de Regional Internet Registry (RIR) da IANA.



Fonte: IANA (2020).

Um endereço IPv4 é um número de 32 bits, sendo representado na forma *a.b.c.d/x*, em que x é um número inteiro entre 0 e 32 que indica o tamanho do prefixo da rede, o qual representa a quantidade de bits da máscara da rede. Por exemplo, uma máscara 255.255.255.0 é representada por /24. Os $(32-x)$ bits restantes são os bits dos *hosts*. Essa representação é denominada CIDR (*Classless Inter-Domain Routing*), uma representação sem classes, adotada após o endereçamento com classes (A, B, C, D e E) cair em desuso devido ao desperdício na alocação de endereços.

Cálculos de VLSM operam sobre o CIDR e permitem a criação de sub-redes dentro de uma sub-rede (recursivamente). Por exemplo, um ISP tem uma rede /21 alocada pelo NIC.br e pode dividi-la em sua intranet em 8 sub-redes /24, ou em 4 sub-redes /23 ou até mesmo em 2 sub-redes /23 mais 4 sub-redes /24 de maneira mista.

2.1.1 Mitigando a escassez do IPv4 com CGNAT

O número de dispositivos conectados à internet tem crescido de tal forma que ultrapassou a capacidade de atendimento de IPs que um ISP pode oferecer aos seus assinantes. Para contornar isso, a estratégia adotada é, a princípio, oferecer um IP público na interface de saída do roteador de borda (WAN) da sub-rede do cliente e uma faixa de endereços privados na rede

local (LAN) do mesmo, sendo feita a tradução de endereços na interface WAN/LAN. IPs privados são blocos definidos pela RFC 1918 e que não devem ser anunciados na internet, sendo restritos ao uso em intranets para o funcionamento do artifício do NAT (*Network Address Translation*).

Existem ISPs que não possuem endereços o suficiente para todos os assinantes, sendo necessário recorrer ao recurso do CGNAT (*Carrier-grade* NAT) para contornar o problema. O CGNAT implementa uma camada de NAT na WAN do cliente, deixando de entregar um IP público para alocar um IP privado ao gateway cliente. IPs privados de CGNAT são definidos pela RFC 6598 e é através deles que é implantado o NAT da operadora, em que um mesmo endereço público é compartilhado por vários assinantes.

O NAT funciona porque é a camada de transporte a responsável em estabelecer comunicação entre dois hosts e não a camada de rede, isto é, uma porta definida pelos respectivos sistemas operacionais dos hosts garante a conexão fim-a-fim na internet. Assim, a função do IP é rotear os pacotes e do TCP estabelecer a conexão HTTP, por exemplo. Um cliente com IP público dedicado tem a sua disposição 65535 portas, o que lhe daria a possibilidade de estabelecer, teoricamente ao máximo, 65535 conexões simultâneas.

Como no NAT os clientes finais compartilham um único IP público, todas as portas que estão associadas a esse IP serão distribuídas entre eles pelo roteador de NAT, sendo que para cada solicitação de conexão será inserido na tabela de tradução de endereços o par $(IP_{publico}, Porta_{publica})$ associado com o par $(IP_{privado}, Porta_{privada})$ de forma aleatória e não conflituosa caso o par já exista na tabela, dado um tempo de vida para esse vínculo. Essa técnica tradicional de NAT é conhecida como *masquerade*, por mascarar toda rede privada atrás do NAT através de um único IP.

2.1.2 Aspectos legais quanto ao uso de NAT

Embora a técnica de NAT *masquerade* resolva o problema da escassez de endereços públicos, existe uma particularidade que não pode ser omitida. De acordo com o Art. 13 do Marco Civil da Internet (Lei nº 12.965/2014), "na provisão de conexão à internet, cabe ao administrador de sistema autônomo respectivo o dever de manter os registros de conexão [...] pelo prazo de 1 (um) ano", sendo que um registro de conexão é definido pelo "conjunto de informações referentes à data e hora de início e término de uma conexão à internet, sua duração e o endereço IP utilizado pelo terminal" (BRASIL, 2014). A lei ainda destaca no Art. 22 que um

juiz pode ordenar ao ISP o fornecimento dos registros de conexão à internet de um determinado cliente com a finalidade de obtenção de provas para processos judiciais.

Dessa forma, o ISP tem a obrigação legal de manter o rastreio sobre qual IP cada um de seus clientes utilizou para navegar na internet, pois caso ele esteja utilizando a rede para cometer algum crime, a polícia conseguirá encontrá-lo. Isso parte do princípio de que na internet todos devem ser identificados pelo IP como sendo seu endereço virtual, porém o NAT *masquerade* quebra essa identidade por não ser determinístico na tradução do endereço. Por isso, um ISP não pode simplesmente resolver a escassez por IPs utilizando uma metodologia de NAT desenvolvida para redes de escritórios domésticos e de pequenas empresas. No ISP, deve ser utilizada, por questões legais, a técnica específica e determinística chamada de CGNAT, também conhecida como NAT de operadora.

2.1.3 Técnicas de CGNAT

Existem duas técnicas adotadas na implantação do CGNAT determinístico: o CGNAT horizontal e o vertical. Ambas garantem a operação do NAT, sendo a diferença entre uma e outra simplesmente a metodologia utilizada no mapeamento entre as redes público-privadas e as faixas de porta.

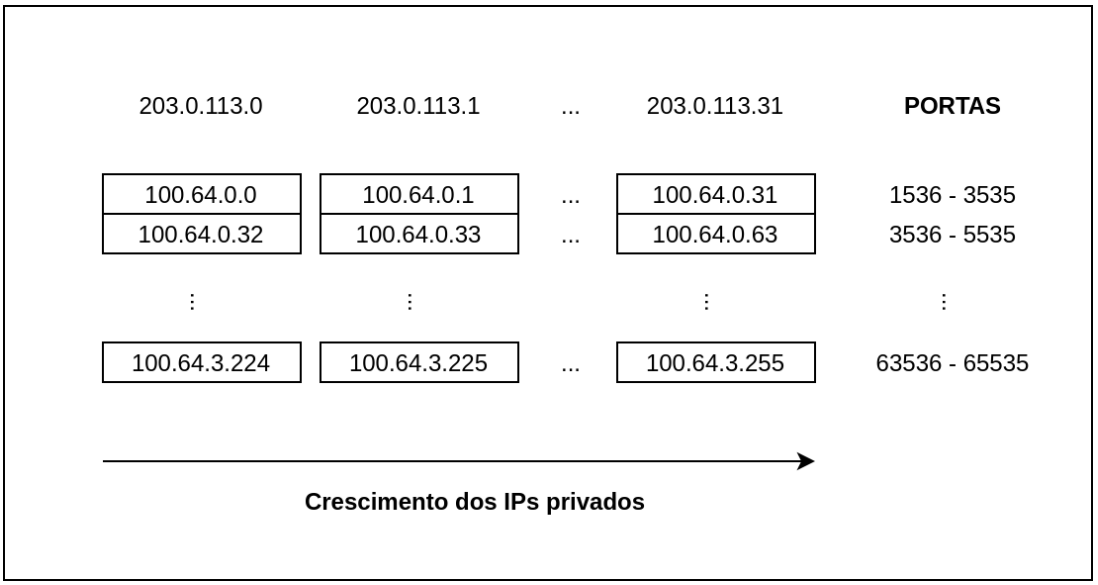
No modelo horizontal, o mapeamento entre os endereços públicos e privados é feito horizontalmente para uma faixa determinada de portas, como é ilustrado na Figura 2.2, em que o sentido de crescimento numérico dos endereços privados acompanha os endereços públicos. Essa técnica possibilita a implantação simplificada por uso de ferramentas de *netmapping*.

No modelo vertical, o sentido de crescimento dos IPs privados não coincide com os IPs públicos, pois o sentido desta vez segue o crescimento do *range* de portas. Assim, é tomado um IP público como referência e são mapeadas as faixas de portas para os IPs privados consecutivamente, como pode ser visto na Figura 2.3. Essa técnica é de mais fácil entendimento, porém requer mais trabalho para implantação por demandar criação de várias regras individuais de NAT.

2.1.4 Desvantagens do NAT

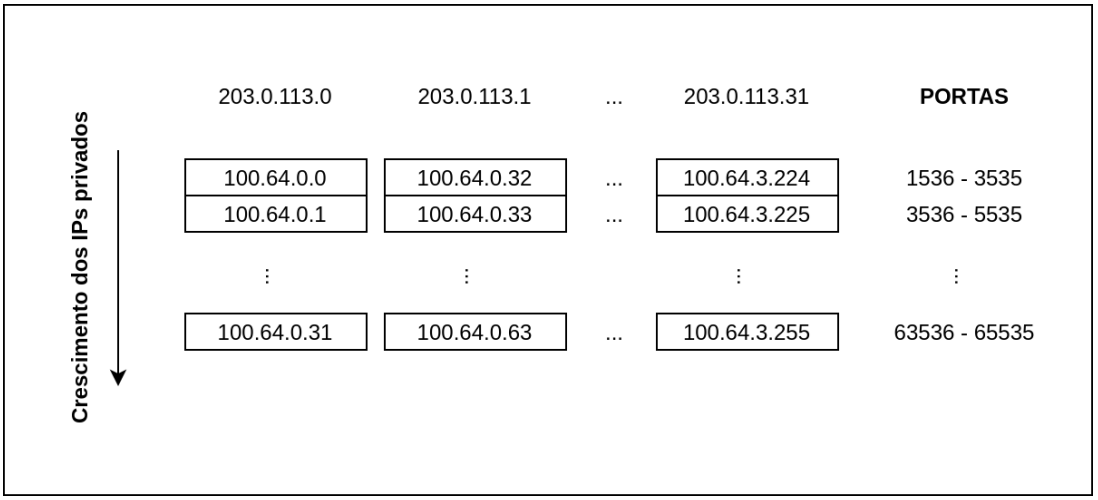
Apesar de solucionar a escassez de endereços de um ISP, o CGNAT implementa uma camada dupla de NAT para os assinantes, o que não permite o funcionamento de redirecionamentos de portas de maneira simples e direta. O redirecionamento de portas nesse cenário,

Figura 2.2 – Ilustração do modelo de CGNAT horizontal.



Fonte: do autor (2020).

Figura 2.3 – Ilustração do modelo de CGNAT vertical.



Fonte: do autor (2020).

necessita da compatibilidade do dispositivo adotado para implantação do CGNAT e fica limitado a uma faixa de portas definida pelo mapeamento.

Como consequência do NAT duplo, protocolos P2P não funcionam, pois não existe comunicação de entrada direta com o *host*, sendo necessário o uso de recursos para travessia de NAT, como reversão de conexão ou repasses para aplicação com um nó intermediário para resolução da limitação imposta (KUROSE; ROSS, 2014).

Do ponto de vista do modelo OSI, o NAT viola o encapsulamento da pilha de protocolos por trabalhar em conjunto nas camadas de rede e de transporte, com a tradução entre IP e porta, sem hierarquia.

2.1.5 Migração para IPv6

Como solução aos problemas estruturais e operacionais do NAT, foi proposta a migração para o IPv6 (IP versão 6), que adota como endereço um número de 128 bits, muito mais do que o suficiente para endereçar unicamente todos os dispositivos em LAN sem o uso de IPs privados. Como todos os dispositivos podem ter um IP público através do IPv6, não é necessário o conceito de NAT com o novo protocolo.

Embora o IPv6 seja a solução, não basta apenas o ISP implantá-lo, é necessário que o acordo de tráfego no novo protocolo também esteja fechado com os servidores de sistemas e de provedores de conteúdo. Enquanto isso não for um tecnologia de ponta-a-ponta, o CGNAT é um modelo satisfatório no processo de transição. Por exemplo, muitos sistemas governamentais ainda utilizam somente IPv4. Dessa forma, os provedores devem manter a conectividade através do protocolo antigo para que não haja prejuízo aos clientes.

Além do mais, o trabalho dos ISPs de pequeno e médio porte demanda atenção em vários problemas operacioanis que a tarefa de implantação do IPv6 acaba perdendo prioridade. Ultimamente, todos os roteadores de *core* já são todos compatíveis com IPv6 nativamente, sendo o impecílio para implantação final do protocolo uma questão de gestão de prioridades.

2.2 Segurança de ambientes de rede

Sistemas em rede estão suscetíveis a ataques, pois a internet é promíscua por sua própria natureza. O objetivo da segurança é minimizar os riscos, pois não existe sistema 100% seguro. A cada dia são descobertas novas falhas e as correções também evoluem de forma constante.

A segurança da informação é regida por quatro pilares: autenticidade, confidencialidade, integridade e disponibilidade. A segurança de redes tem seu foco principal em disponibilidade, prezando pela minimização de vulnerabilidades que possam ser exploradas para causar interrupção no serviço, sejam exploradas através de falhas físicas ou através de ataques de negação de serviço.

Entre as principais ferramentas de defesa disponíveis para segurança de rede, estão o firewall e a VPN (Virtual Private Network), que serão descritas a seguir.

2.2.1 Firewall

Firewall é um ponto entre duas ou mais redes onde é possível controlar todo o tráfego de dados que passa através dele. Assim, esse ponto único constitui um mecanismo utilizado para proteger uma rede confiável de uma rede pública não-confiável (NAKAMURA; GEUS, 2007).

As técnicas básicas de firewall são a filtragem de pacotes estática (*stateless*) e a filtragem de pacotes baseada em estados (*stateful*). No firewall *stateless*, as regras são aplicadas na camada de rede (camada 3) e de transporte (camada 4), sendo feita a filtragem a partir das informações de IP, porta e protocolo contidas no cabeçalho do pacote. A técnica *stateless* apresenta alto desempenho e performance para gerenciamento de tráfego devido a sua simplicidade. Em contrapartida, não é uma técnica suficiente para filtragem de serviços de portas dinâmicas, em que as portas de comunicação são definidas em tempo de execução, como nos protocolos RPC e FTP (NAKAMURA; GEUS, 2007). O problema das portas dinâmicas é possível de ser tratado com um firewall *stateful*, que mantém registros das conexões para filtragens mais inteligentes e dinâmicas, tendo como custo um maior consumo de recursos de processamento.

As vulnerabilidade de obtenção de informação, em que atacantes oportunistas podem conseguir informações em banners de protocolos de rede local expostos na internet, podendo conseguir informações valiosas para auxiliar em um ataque de invasão, podem ser tratadas com firewall pelo uso de regras de acesso definidas por IP. Além disso, a vulnerabilidade de invasão também pode ser corrigida com um firewall fazendo bloqueio de portas aos invasores.

O bloqueio de portas também auxilia na minimização de ataques de negação de serviço, pois um atacante pode conseguir amplificar o tráfego e congestionar a rede explorando protocolos como DNS e SSDP, que têm o pacote de resposta muito maior que a requisição, podendo inundar a rede em um ataque coordenado. Com as portas desses serviços fechadas àqueles que não necessitam acessá-las, mais segurança é agregada à rede.

O firewall não é a solução total para segurança da rede. É fundamental selecionar os usuários que podem acessar a rede e definir os seus respectivos níveis de acesso (somente leitura ou leitura e escrita). A autenticação e a autorização são também importantes aspectos a serem implementados na infraestrutura de segurança (NAKAMURA; GEUS, 2007).

2.2.2 VPN

Uma VPN tem como funcionalidade prover conexão a uma rede local através da internet por meio de um túnel de conexão criptografado. O mecanismo da VPN provê serviço de acesso

remoto como também serve de mecanismo para controle de acesso, fornecendo uma camada de autenticação à rede interna e de autorização quando combinada com um firewall.

Os conceitos que fundamentam a VPN são a criptografia e o tunelamento. A criptografia é utilizada para garantir a autenticidade, o sigilo e a integridade dos dados através da conexão, e é a base da segurança dos túneis VPN (NAKAMURA; GEUS, 2007). Uma VPN implantada com protocolo PPTP ou L2TP garante somente autenticidade, enquanto IPSec e protocolos baseados em TLS (como Stunnel e aplicações modernas de VPN) somam integridade e confidencialidade ao túnel.

2.3 Topologia de um ISP

O modelo comum de topologia adotado por operadoras de internet é formado por um triângulo entre roteadores B-RAS (*broadband remote access server*), CGNAT e borda, conforme Figura 2.4. Os clientes finais conectam-se à rede através de PPPoE ao roteador B-RAS, que por sua vez encaminha o tráfego de IP público à borda que será roteada para a internet. Tráfego de IP privado de CGNAT (RFC 6598) é encaminhado ao equipamento que traduz os IPs privados em públicos antes de serem roteados à internet pela borda.

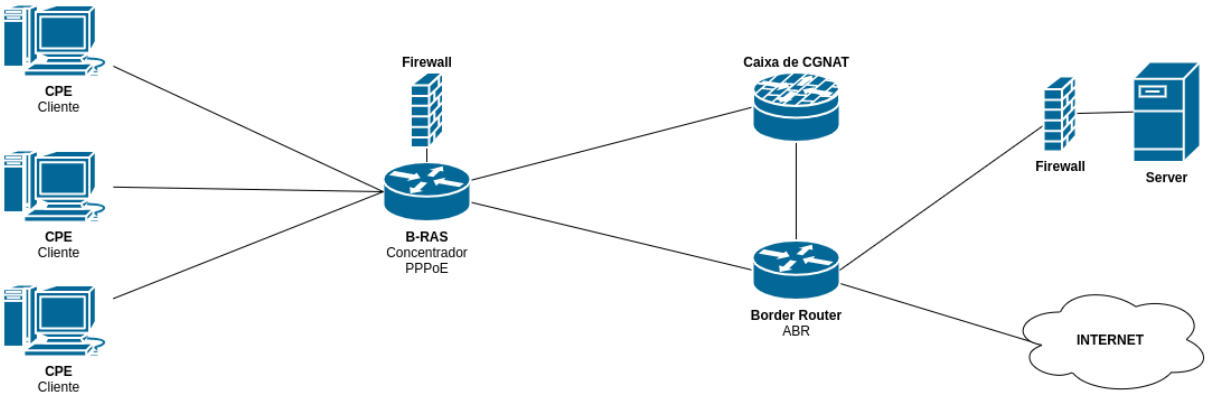
Dependendo da dimensão da rede, é possível adotar modelos simplificados para economia de recursos, fazendo a mescla das funções de um ou de mais equipamentos em um único, desde que o equipamento suporte essa configuração.

A Figura 2.4 é uma representação genérica de camada 3 do núcleo da rede de um ISP. Existem muitos outros elementos de conexão de camada 2 e de camada 1 (lógica e física respectivamente) entre os dispositivos que não estão na imagem, como enlaces de rádio ou de fibra óptica. Também foi omitido, por simplificação, que dentro do salto entre o ABR e a Internet na Figura 2.4, existe o ASBR (*Autonomous System Border Router*), que é resposável por rotear o tráfego para a internet.

Na infraestrutura lógica do ISP, também entram os servidores de serviços locais, que provêm servidores de autenticação RADIUS, serviço de DNS local para respostas mais rápidas, Proxy HTTP[1] para poupar consumo de banda de acesso à internet e serviço de VPN.

[1] Também conhecido como CDN – *Content Distribution Network*.

Figura 2.4 – Modelo de topologia de um ISP.



Fonte: do autor (2020).

3 ENDEREÇAMENTO IP

O endereço IP é a identificação que possibilita o tráfego de dados desde sua origem até o destino através da rede mundial de computadores, por isso é necessário fornecer endereços IP para todos os dispositivos que se conectam à internet. Em um provedor, a rede (que utiliza um conjunto de IPs com mesmo prefixo) é dimensionada de acordo com a quantidade de clientes que serão atendidos dentro de uma determinada área de abrangência, que geralmente é atomizada por cidades.

3.1 Rede IP da Minasnet

Na Minasnet, a rede é subdividida em áreas OSPF, sendo que cada área é associada a uma cidade. Zonas rurais e vilarejos recebem o mesmo código de área da cidade a qual pertencem. Assim, a sub-rede é dimensionada de acordo com o tamanho da cidade e com a quantidade de clientes associados àquela área. Geralmente, um bloco /24 de endereços é o menor espaço de endereçamento atribuído a uma área e um bloco /22 é o maior. Por exemplo, a franquia de Perdões tem uma rede /22, isso significa que existem 1024 endereços públicos dedicados para atendimento aos clientes dessa cidade.

Os IPs são associados aos CPEs (dispositivo *gateway* da rede interna de um cliente) através de um túnel PPPoE fechado com o concentrador (B-RAS). Como o próprio nome sugere, o concentrador centraliza todas as conexões de camada 3 dos clientes conectados em um único equipamento. Entretanto, dependendo do hardware e da configuração do software de um equipamento concentrador, pode ser necessário a utilização de mais de um equipamento para dividir a carga do servidor PPPoE.

Na Minasnet, um exemplo disso foi a franquia de Oliveira, uma das mais recentemente atendidas pelo ISP. A princípio existia um único concentrador PPPoE que, devido à demanda de clientes entrantes, teve sua carga dividida com um segundo concentrador instalado junto a ele. O padrão atual da empresa é manter no máximo 1024 clientes em um único concentrador.

Para documentar a rede IP do AS (conjunto de todas as sub-redes públicas de um ISP), é utilizado o software PHPIPAM[1]. Neste software é possível criar o aninhamento entre sub-redes e deixar descrito qual a finalidade de cada uma delas, facilitando consultas e manipulações de VLSM nessas redes dentro do próprio sistema. Tomando o exemplo de Perdões, no PHPIPAM

[1] PHPIPAM <https://phpipam.net> é um software *open-source* de gerenciamento de endereços de rede (IP, VLAN e etc.).

existe uma rede com mesmo nome da cidade, na qual estão documentados todos os IPs públicos da rede /22 dimensionada para a mesma, além dos IPs privados utilizados na franquia, que estão contidos nas redes 10.0.0.0/8, 172.16.0.0/12 e 192.168.0.0/16, definidas pela RFC 1918 (MOSKOWITZ et al., 1996), e o espaço compartilhado 100.64.0.0/10 definido pela RFC 6598 (WEIL et al., 2012). Dentro da rede IP pública, é reservada uma sub-rede para alocação de IP fixo e o restante dos IPs são alocados dinamicamente para os clientes.

Dado que a franquia de Perdões possui muito mais do que 1024 clientes, um *pool*[2] de prefixo /22 não atenderia à demanda. A solução para a escassez de endereços na franquia é a utilização do artifício do NAT, disponibilizando IPs privados aos clientes e fazendo associação do par IP-porta com o IP público de destino do NAT. Redes privadas não devem ser anunciadas na internet pública e a única forma de comunicarem-se com o mundo é através do NAT.

3.2 Metodologia para implementação de CGNAT

Observando os requisitos definidos pelo Art. 13 da Lei nº 12.965/2014, descritos na Seção 2.1.2, e relacionando-os com a infraestrutura de um concentrador, devemos registrar o *timestamp* de início e de fim da sessão PPPoE do cliente bem como qual o IP foi disponibilizado a ele durante essa sessão. A sessão PPPoE inicia-se no instante em que o cliente estabelece conexão e se encerra quando acontece a desconexão. Salvar essas informações é simples quando se utiliza um servidor RADIUS[3], pois a função básica desse serviço é autorizar a conexão de assinantes, atribuindo IP e armazenando o *log* das informações de acesso.

A RFC 6888 estabelece requisitos para implementação do CGNAT de maneira segura. Um ponto importante dessa RFC é a definição obrigatória do mapeamento direto entre os *pools* de endereços públicos e privado, imprescindível para que o CGNAT seja determinístico. Outro ponto importante é a manutenção de registro das informações do assinante conectado, assim como especificado pelo Marco Civil da Internet.

Embora a RFC 6888 esteja em consonância com o Marco Civil da Internet, existe divergência. A diferença é que o Marco Civil estabelece normas tipicamente orientadas para provedores que entregam somente IP público para seus clientes, por não estabelecer uma regra de registro de número de porta nas conexões. A RFC 6888 é mais ampla por contemplar as

[2] Diferente de uma rede, que tem reservado o primeiro e o último endereço para o *host* e para o *broadcast* respectivamente, no *pool* todos os endereços são alocados, inclusive 0 e 255.

[3] A Minasnet utiliza o FreeRADIUS <https://freeradius.org>.

necessidades de registro de conexões através de CGNAT para garantir o rastreio da identidade dos clientes na internet, definindo como parâmetros para *log* (PERREAULT et al., 2013):

a) o protocolo de transporte;

b) o IP interno;

c) o IP externo de origem;

d) a porta externa de origem;

e) o *timestamp* de registro.

Vale ressaltar que IP externo de origem e porta externa de origem são valores do lado do provedor, sendo que IP de destino e porta de destino seriam do lado da aplicação. Não é recomendado armazenar informações de destino dos pacotes, uma vez que isso quebraria a privacidade de navegação dos assinantes por rastrear tudo que eles tem acessado na internet (PERREAULT et al., 2013).

O problema de se seguir à risca os cinco parâmetros definidos acima é que demandaria muito espaço em disco para a manutenção do rastreio das conexões, pois seria necessário armazenar no banco de dados de *log* cada novo registro na tabela de tradução de endereços. Seria um volume tão grande de dados que até uma consulta para atender a uma solicitação judicial poderia ser demorada, lembrando que o banco de dados deve manter por 1 ano todas as informações.

A solução desse problema é utilizar técnicas que fazem mapeamento direto entre uma faixa contígua de portas, de tamanho padronizado, para os IPs públicos e privados. Com o mapeamento de portas, só é necessário registrar em *log* o IP interno do cliente e os *timestamps* de início e de fim da sessão PPPoE, resultando em uma redução considerável no volume de dados, pois serão somente três campos registrados ao invés dos cinco.

Outra melhoria para redução do volume de dados é utilizar dois campos de *timestamp* ao invés de um só – um para o momento de conexão e outro para o de desconexão. Assim, a inserção de novas linhas na base de dados será necessária somente quado o cliente conectar, inserindo o *timestamp* no campo de conexão e deixando o de desconexão *null* enquanto o cliente estiver conectado. No momento da desconexão, basta atualizar o campo *null* com a data e hora da desconexão.

Apesar de o mapeamento de portas não gerar alto volume de dados de *log* de conexão dos assinantes, para que seja uma técnica suficiente na implementação do CGNAT é preciso manter documentadas as regras de mapeamento de IP-porta. A utilização de uma planilha é a maneira mais prática para realizar tal tarefa. Na Minasnet, é mantida uma planilha na qual as informações são registradas de acordo com colunas contendo:

a) nome do concentrador alvo do CGNAT;

b) nome da franquia;

c) sub-rede interna;

d) sub-rede externa;

e) *timestamp* de início de vigência da regra.

f) anotações.

Os dois primeiros itens são apenas por questão de organização, pois existem dezenas de concentradores no ISP e todos estão documentados nessa planilha, sendo somente as informações de mapeamento de redes interna e externa e o *timestamp* relevantes para eficácia do registro. As anotações tem informações sobre desativação da regra de CGNAT ou modificações nas redes, não sendo criado mais campos específicos porque o objetivo é que seja alterado o mínimo possível. Até hoje, poucas vezes houve alterações no mapeamento das sub-redes. O Quadro 3.1 exemplifica o uso da planilha para controle e documentação dos mapeamentos.

Quadro 3.1 – Exemplo de planilha para controle de CGNAT.

Concentrador	Franquia	Rede interna	Rede externa	Início	Anotações
CON-PER-01	Perdões	100.64.0.0/22	203.0.113.0/27	2019-04-18 17:15	Vigente
CON-PER-02	Perdões	100.64.4.0/22	203.0.113.32/27	2019-05-02 12:05	Vigente
CON-PER-03	Perdões	100.64.8.0/22	203.0.113.64/27	2019-11-13 16:35	Vigente

Fonte: do autor (2020).

A Minasnet adota por padrão a disponibilização de 2.000 portas para cada cliente atrás do CGNAT, definindo o intervalo da faixa entre 1536 e 65535. Isso significa que, para cada IP público do CGNAT do ISP, existem 32 clientes internos conectados à internet através dele. Esse dimensionamento de 1 IP externo para 32 internos denomina a razão de compartilhamento

1:32. As portas de uso reservado (0-1023) não são usadas e a numeração começa a partir de 1536 por questões de arredondamento de cálculos.

Os cálculos a seguir demonstram o dimensionamento do mapeamento de portas descrito. O primeiro passo é verificar a quantidade de portas Δ que estão sendo dedicadas ao CGNAT por um único IP, simplesmente subtraindo os limites de portas definidos e somando 1, pois a primeira porta também é contabilizada:

$$\Delta = porta_{maior} - porta_{menor} + 1 = 65535 - 1536 + 1 = 64000 \tag{3.1}$$

O que resulta em 64.000 portas. Como cada IP interno tem 2.000 portas mapeadas para ele, então a quantidade de IPs internos n para cada IP público será:

$$n = \frac{64000}{2000} = 32 \tag{3.2}$$

Como dito anteriormente, um concentrador da Minasnet normalmente é dimensionado para atender até 1024 clientes. De acordo com a proporção 1:32, são necessários 1024 IPs internos e 32 IPs externos para implementação do CGNAT nesse caso. A quantidade de IPs externos é obtida tirando a razão 1024 por 32, pois a razão de compartilhamento dada é 1:32, o que resulta em 32 IPs. Então, nesse concentrador deve ser criado uma mapeamento de uma rede interna com 1024 endereços (/22) para uma rede externa de 32 endereços (/27).

Calcular mapeamentos para outros prefixos de rede é simples, pois seguem a mesma lógica usada no mapeamento /22 entre /27. O Quadro 3.2 mostra alguns dos mapeamentos que são possíveis de serem feitos seguindo a metodologia usada aqui. A demonstração pode ser feita alterando os valores correspondentes dos cálculos supracitados, em sempre será obtida a razão constante de 1:32 entre IP externo e interno.

3.2.1 Implementação de CGNAT no RouterOS

Até aqui, foi fundamentada a metodologia utilizada para o CGNAT. Nesta seção, será apresentada a implementação das regras em roteadores MikroTik através do sistema operacional embarcado RouterOS[4], com geração das regras através de ferramenta desenvolvida em Python de autoria própria.

[4] RouterOS <https://mikrotik.com/software> é o sistema operacional nativo dos equipamentos MikroTik, sendo um software proprietário com *kernel* Linux.

Quadro 3.2 – Mapeamento direto entre sub-redes internas/externas usando CGNAT 1:32.

Prefixo privado	Prefixo público	Quantidade de clientes
...	...	...
/20	/25	4096
/21	/26	2048
/22	/27	1024
/23	/28	512
/24	/29	256
/25	/30	128
/26	/31	64
/27	/32	32

Fonte: do autor (2020)

O primeiro passo para o desenvolvimento prático do CGNAT é entender o que o RouterOS nos proporciona para atingir esse objetivo. Por ser baseado em Linux, a ideia básica é construir as regras de NAT através do módulo de firewall do sistema, pois o NAT é interpretado como uma regra de firewall por fazer modificação no cabeçalho dos pacotes de dados ao alterar os valores de endereço e de porta originais.

O módulo utilizado é a tabela NAT do firewall do RouterOS, sendo necessário configurar os seguintes parâmetros na geração das regras conforme definidos na documentação (MIKROTIK. . . , 2020a):

a) `action`, especifica a ação que deve ser executada; neste caso, a ação definida foi o `netmap`, que consiste no mapeamento direto entre as redes interna e externa;

b) `chain`, é configurado como `srcnat`, pois os pacotes alvos da regra originam-se na rede interna ao firewall;

c) `protocol`, são configurados tanto TCP e UDP individualmente;

d) `src-address`, é a sub-rede interna;

e) `to-addresses`, é a sub-rede externa;

f) `to-ports`, é a faixa de portas do IP público alocada.

A única ressalva é que, para o pleno funcionamento do NAT para pacotes de protocolos de camada 3 (ICMP), devem ser criadas regras em que não estejam definidos os itens (c) e (f).

Com os parâmetros definidos, o template do comando a ser executado na CLI é dado conforme o exemplo da Figura 3.1. O método de utilização simplificada do `netmap` foi obtido de

(MAIA, 2018) e é uma técnica de CGNAT horizontal, ilustrada na Figura 2.2. Neste exemplo, são utilizadas as sub-redes indicadas no exemplo do CON-PER-01 no Quadro 3.1. Pode-se notar que foi utilizado um prefixo /27 ao invés de um /22 no comando, isso acontece porque para que a regra de `netmap` faça o mapeamento direto, as redes devem ter o mesmo prefixo para que a correspondência entre os IPs seja possível. Assim, uma rede /22 deve ser subdividida em 32 sub-redes /27 para que seja feito o mapeamento com uma rede pública de prefixo /27. Para fazer o casamento entre outros valores de prefixos, o princípio é o mesmo.

Figura 3.1 – Template do comando de configuração de netmap no RouterOS.

```
/ip firewall nat
    add action=netmap           \
    chain=srcnat                \
    protocol=tcp                \
    src-address=100.64.0.0/27   \
    to-addresses=203.0.113.0/27 \
    to-ports=1536-3535          \
    disabled=yes
```

Fonte: do autor (2020).

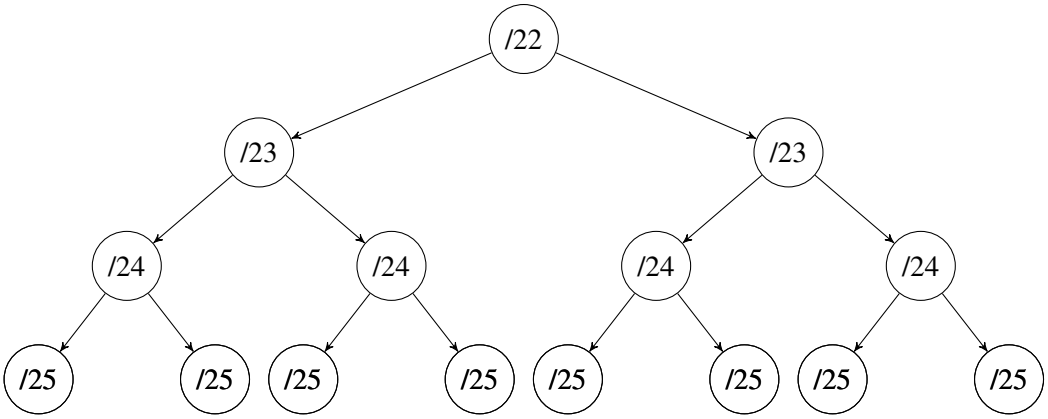
Acontece que, fazer geração manual de todas essas regras é um trabalho extenso e cansativo, que pode ficar sujeito a falhas humanas. Para isso, foi desenvolvido um utilitário de CLI para geração das regras para construir `netmap` entre qualquer prefixo, desde que respeite a razão 1:32. O software foi desenvolvido em Pyhton.

O fundamento utilizado na construção do programa baseia-se na geração de uma árvore binária oriunda das subdivisões recursivas de uma rede privada de determinado prefixo, situada na raiz, até que chegue em uma camada em que as folhas tenham o mesmo prefixo que a rede pública, procedimento ilustrado na Figura 3.2. A partir daí, o algoritmo obtém uma lista com todas as sub-redes formadas pelas folhas dessa árvore, e gera as regras de `netmap` conforme definido no template da Figura 3.1, incrementando o número de portas em um valor fixo de 2.000.

A ferramenta desenvolvida foi nomeada como `py-cgnat`. O código-fonte está disponível no GitHub[5] do autor e o executável pode ser baixado através do repositório oficial do Python[6]. O software foi desenvolvido seguindo as convenções do ecossistema Python e é

[5] <https://github.com/williamabreu/py-cgnat>.
[6] <https://pypi.org/project/pycgnat/>.

Figura 3.2 – Árvore binária das subdivisões recursivas de um /22 até um /25.



Fonte: do autor (2020).

open–source, sendo lançado sob licença permissiva MIT para garantir liberdade de uso e de distribuição sem complicações. A versão 1.0b1 é a utilizada neste trabalho.

3.2.2 Utilização do utilitário py-cgnat

A seguir, será utilizado o mapeamento do concentrador CON-PER-01 (conforme Quadro 3.1) para exemplificar o uso do programa `py-cgnat` via terminal de comando. Para fazer a geração das regras de `netmap` para RouterOS, o comando a ser executado no terminal segue conforme mostrado na Figura 3.3. Devem ser informados os seguintes parâmetros: a rede privada e a pública, a opção `gen` para acionar o módulo de geração e a plataforma onde será configurado o CGNAT, que neste caso é o RouterOS. O último parâmetro é opcional e é o nome do arquivo de destino das regras caso queira ser salvo, pois se deixado em branco, as regras serão impressas no próprio terminal de comando. O campo plataforma foi posto por questão de extensibilidade, para deixar o software pronto para futuras novas versões, em que se pretende suportar equipamentos de outros fabricantes.

Figura 3.3 – Exemplo de uso do programa py-cgnat para geração de CGNAT para RouterOS.

```
In: pycgnat 100.64.0.0/22 203.0.113.0/27 gen routeros rules.rsc
Out: null
```

Fonte: do autor (2020).

O programa também calcula as traduções entre IP público-privado através do módulo `trans`. Um exemplo é uma consulta para o IP privado 100.64.0.47, que retorna o IP público 203.0.113.15 e a faixa de portas definida entre 3536 e 5535 com o comando da Figura 3.4. De

maneira reversa, uma consulta para o endereço 203.0.113.15 na porta 5000 retorna o cliente com IP 100.64.0.47, que pode ser verificada conforme Figura 3.5. As consultas de traduções independem de plataforma, pois fazem parte da lógica da metologia de mapeamento de portas utilizada.

Figura 3.4 – Exemplo de uso do programa py-cgnat para traduções de IP privado para público.

```
In: pycgnat 100.64.0.0/22 203.0.113.0/27 trans -d 100.64.0.47
Out: {"public_ip": "203.0.113.15", "port_range": [3536, 5535]}
```

Fonte: do autor (2020).

Figura 3.5 – Exemplo de uso do programa py-cgnat para tradução de IP público para privado.

```
In: pycgnat 100.64.0.0/22 203.0.113.0/27 trans -r 203.0.113.15:5000
Out: {"private_ip": "100.64.0.47", "port_range": [3536, 5535]}
```

Fonte: do autor (2020).

Além disso, o software foi desenvolvido para também ser utilizado como biblioteca de programação, com o objetivo de automatizar processos utilizando Python ou de integrar com sistemas já existentes, caso haja necessidade. Uma breve documentação (em inglês) do uso pode ser obtida no Apêndice A ou nos repositórios do `py-cgnat` no GitHub ou no PyPI.

3.3 Considerações sobre o uso de CGNAT

Uma ressalva para o processo de consulta das traduções de endereços é que, caso seja necessário descobrir qual cliente estava navegando com determinado IP em um determinado instante, é necessário consultar no banco do RADIUS qual o *login* PPPoE esteve com aquele IP no dado momento. Esse tipo de solicitação acontece geralmente por ordem judicial ao ISP, a fim de descobrir a identidade de criminosos, que atuam na internet ou através dela.

É muito importante seguir a documentação exemplificada na planilha do Quadro 3.1 para garantir o cumprimento do Marco Civil da Internet. Por isso, a utilização de faixas de endereços distintas para clientes com dívida, denominado *pool* de bloqueio, é considerada uma má prática quando não aplicada a técnica de CGNAT em clientes com velocidade de navegação reduzida, porque no período de bloqueio por dívida, seria impossível fazer rastreamento por IP.

3.3.1 Dificuldades encontradas na implantação

A dificuldade encontrada no *deploy* do CGNAT foi que dois concentradores não estavam com a tabela NAT do firewall funcionando, sendo que o motivo da falha era desconhecido pela equipe. A solução foi subir as regras no dispositivo do ABR, pois como eram poucos clientes, as regras da rede do CGNAT não impactou o desempenho do roteador de borda, sendo feito acompanhamento de consumo de CPU durante momentos de pico do dia (por volta de 20h) para constatar isso. Outra fonte de constatação foi a ausência de reclamações por clientes daquela área onde foi aplicada essa solução de contorno. Também é possível colocar roteadores MikroTik dedicados ao CGNAT intermediando concentrador e borda, mas não foi adotada esta técnica como finalidade do trabalho.

4 SEGURANÇA DE REDE

Estar conectado à internet requer cuidados, pois dispositivos expostos à rede pública estão sujeitos às ameças de segurança que podem atacá-los. Sistema completamente seguro não existe, mas usuários finais devem tomar os devidos cuidados em suas redes locais bem como os ISPs também devem tomar as medidas que estiverem ao alcance. No ISP, as competências de segurança atribuídas ao NOC são manter seguros os serviços, protocolos e as redes de telecomunicações, que estão nos níveis mais baixos da implementação das políticas de segurança da informação, enquanto o desenvolvimento de software e interação humano-computador estão em alto nível. A segurança é um acordo que deve ser cumprido por todas as partes em todos os níveis para minimização dos incidentes.

4.1 IP Blacklist

A Minasnet, como sendo um AS, tem delegação sobre vários blocos de endereços IP que dão aos seus assinantes identidade na internet. O primeiro problema quanto à segurança surge neste ponto, pois a conectividade traz riscos, que se iniciam pelo IP público através do qual os ataques conseguem propagar-se pela internet (ou pela intranet, via rede privada).

Grande parte dos *worms* são propagados pela internet através de *spam*, que são enviados de maneira massiva por dispositivos infectados, sejam computadores, smartphones, roteadores domésticos e até roteadores de *core*, basta ter conexão à internet que se torna vulnerável a *botnets*.

Uma forma de controlar o envio de spam, por parte dos provedores de serviço, é a utilização de listas negras com IPs que foram identificados como fonte de envio de *spam*. Chamadas de *Real-time Blackhole List* (RBL), doravante *blacklist*, proporcionam a listagem de IPs detectados recentemente como participantes ativos de *botnet* para envio de *spam*, possibilitando ao provedor de serviço bloquear todo o tráfego de e-mail a partir daquele endereço IP listado.

Neste trabalho, foi feito uso de RBL com a finalidade de monitoramento de IPs do AS que estão listados na blacklist, afim de identificar infecções no núcleo da rede (devido às vulnerabilidades associadas a equipamentos MikroTik com IP público), bem como nos assinantes. Para isso, foi utilizado o *Composite Blocking List* (CBL)[1], uma RBL com consulta no padrão DNS Blacklist (DNSBL).

[1] CBL <https://www.abuseat.org> é uma divisão da Spamhaus.

DNSBL é especificado pela RFC 5782, funcionando com estrutura semelhante a um DNS reverso. Cada IP listado em uma DNSBL tem um subdomínio correspondente, sendo que cada entrada de subdomínio é criada revertendo os octetos do IP e concatenando com o domínio da DNSBL. Uma consulta à DNSBL retorna no registro `A` o endereço 127.0.0.2 e no registro `TXT` a descrição do motivo pelo qual o IP está listado na *blacklist* (LEVINE, 2010). Quando o IP não está na *blacklist*, a consulta retorna o erro `NXDOMAIN`. Por exemplo, uma consulta para verificar se o IP 192.0.2.99 está listado na CBL seria feita a partir da entrada 99.2.0.192.cbl.abuseat.org e, em caso de listagem positiva, o link informado no registro `TXT` daria um relatório detalhado da origem e qual tipo de infecção foi detectada atrás desse IP. Em terminal de comando Linux, a consulta pode ser feita da seguinte forma, para os registros `A` e `TXT`, respectivamente, conforme Figura 4.1.

Figura 4.1 – Exemplo de consulta à DNSBL CBL via terminal Linux.

```
host -t A 99.2.0.192.cbl.abuseat.org
host -t TXT 99.2.0.192.cbl.abuseat.org
```

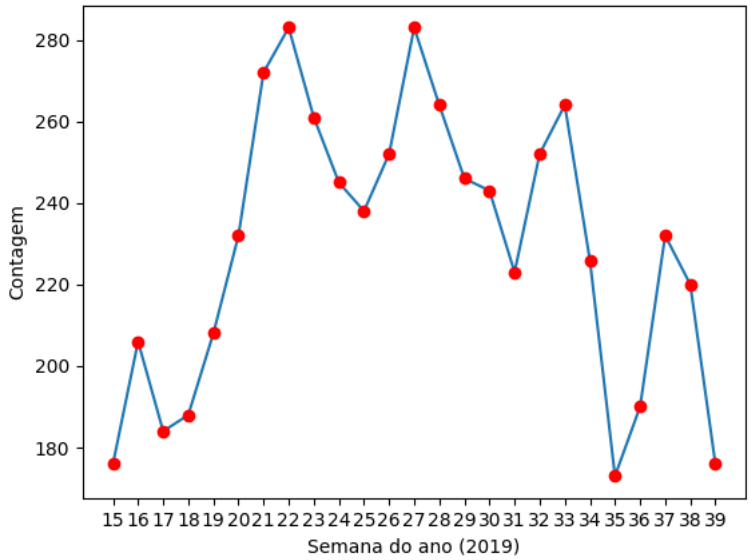
Fonte: do autor (2020).

O monitoramento foi feito de maneira automatizada, através de um simples programa feito em Python para fazer a varredura de todos os blocos de IP do AS, que até o momento é constituído por 8.192 endereços. O programa varre todos os endereços e retorna um resumo contendo somente aqueles que estão listados na *blacklist*, que serão analisados através do relatório detalhado na página da CBL.

O Gráfico 4.1 mostra o resultado do monitoramento de IPs da Minasnet listados na CBL entre abril e setembro de 2019 e o Anexo A é um exemplo de relatório detalhado fornecido pela plataforma, contendo informações de origem, destino e tipo de infecção, além de sugestões para correção do problema. Apesar de o Gráfico 4.1 apresentar subidas e descidas acentuadas na contagem dos IPs listados na *blacklist*, não foi evidenciada nenhuma relação imediata entre a quantidade de IPs listados com as tratativas implementadas em firewall para tentar mitigar o problema, o que caracteriza que o *spam* está originando-se na rede interna dos assinantes e trafegando na internet pela camada de aplicação, como pode ser visto no relatório apresentado no Anexo A, que relata a detecção através de tráfego HTTP.

As informações sobre os endereços IP listados na CBL são detectadas por *honeypots*. Os *Honeypots* são máquinas que emulam determinados sistemas operacionais e serviços, para

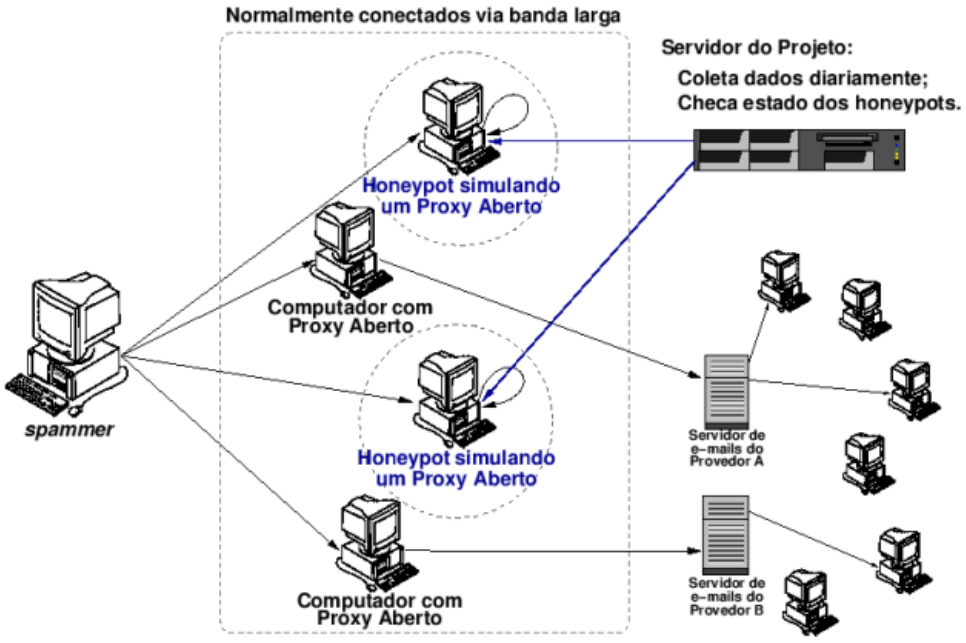
Gráfico 4.1 – Contagem de IPs do AS listados na CBL entre abril e setembro de 2019.



Fonte: do autor (2020).

que um atacante interaja com ela sem que perceba que está entrando em uma armadilha (CERT, 2007). É utilizando este artifício que a CBL faz a detecção de *botnets* e registra o IP de origem do *spam*. A Figura 4.2 mostra um exemplo de arquitetura utilizada para essa detecção. A descrição da metodologia utilizada pela plataforma pode ser obtida (em inglês) no site da CBL.

Figura 4.2 – Arquitetura de um honeypot para detecção de spam.



Fonte: CERT (2007).

A CBL oferece a opção de remoção manual de um IP listado, porém essa ação não soluciona o problema de fato, uma vez que caso a infecção persista, muito provavelmente o IP

retornará à listagem. Por isso, a melhor solução para o problema da *blacklist* é a correção efetiva de vulnerabilidades existentes na rede, sendo que após um período de 28 dias sem nenhuma ocorrência, automaticamente o IP é removido da lista.

4.2 Vulnerabilidades de rede

O CERT.br (Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil) é responsável por tratar incidentes de segurança computacional envolvendo redes conectadas à internet brasileira. Por isso, possui rotina de analisar, por amostragem, IPs registrados pelo Registro.br e notificar os responsáveis pelo AS no qual foi detectada alguma vulnerabilidade.

Vulnerabilidades de rede podem ser exploradas através de portas abertas em dispositivos. Portas abertas de serviços mal configurados são fonte para ataques de invasão, envio de *spam* e DDoS (negação de serviço).

Dessa forma, o CERT.br faz, regularmente, amostragem de IPs dos ASs brasileiros para realizar varredura de portas abertas, filtrando portas de serviços específicos que possam ser explorados por atacantes na internet e notificando as operadoras responsáveis. Uma varredura de portas, ou *scan* de portas, pode ser feita a partir de terminal de comando pela ferramenta `nmap`. Através do comando apresentado na Figura 4.3, pode ser feita a varredura de todas as portas do *host* 192.0.2.99, por exemplo. Os parâmetros `-sU` e `-sT` especificam a utilização de portas UDP e TCP na varredura, respectivamente, `-p-` é um *alias* para `-p0-65535`, que faz varredura de todas as portas possíveis e `-T5` é utilizado para uma execução mais rápida.

Figura 4.3 – Exemplo de uso do nmap para varredura completa de portas.

```
nmap -sU -sT -p- 192.0.2.99 -T5
```

Fonte: do autor (2020).

Quando portas de serviços vulneráveis são detectadas como abertas, o CERT.br envia um e-mail contendo relatório detalhado da vulnerabilidade ao responsável pela administração daquele IP escaneado, além de fornecer sugestões para correção do problema. O Anexo B é um e-mail enviado pelo CERT.br à Minasnet após detectar vulnerabilidade no protocolo SOCKS em dispositivos MikroTik na rede.

A maioria das vulnerabilidades surgem devido a configuração errônea de CPEs por parte dos instaladores nas residências ou nos estabelecimentos comerciais, ou por parte dos analistas

de rede na configuração de roteadores do núcleo da rede. A correção dessas vulnerabilidades pode ser resolvida com a configuração correta de todos os equipamentos, o que é uma meta difícil de ser atingida, devido ao fator humano ser responsável pela garantia das configurações. Assim, quanto menor a dependência de fatores humanos, melhor a abordagem de tratamento de vulnerabilidades. Neste trabalho, foi utilizada a abordagem de firewall para filtragem de portas e VPN para controlar o acesso.

As principais vulnerabilidades exploradas por atacantes que foram tratadas através do firewall implementado são os ataques por obtenção de informações, por invasão e por negação de serviço, que geralmente estão nas notificações do CERT.br por serem recorrentes.

4.3 Implantação de VPN

A abordagem utilizada no trabalho foi, primeiro, controlar o acesso remoto implementando um serviço de autenticação através de VPN e, depois, fazer a filtragem de portas. Para controle de acesso, foi adotado o OpenVPN[2], uma solução gratuita, customizável e portável para várias plataformas. OpenVPN é um serviço completo de VPN SSL que é executável em um servidor Linux, que pode ser integrado com vários outros serviços e protocolos como for desejável. VPNs L2TP/IPSec e PPTP não possuem tantos recursos e flexibilidade como o OpenVPN suporta. Além disso, OpenVPN é um projeto open source, que conta com uma grande comunidade, com boa documentação e com fórum de suporte e de discussão. OpenVPN também possui serviços prontos para empresas e para usuários finais, a partir do pagamento de assinaturas.

O projeto da VPN é bem simples, possuindo apenas dois requisitos: deve ser instalada em um servidor e possuir autenticação de dois fatores (2FA). Assim, o *deploy* do servidor foi feito em uma VPS (*virtual private server*) Debian Linux, mantida no datacenter da Minasnet, com as seguintes configurações de hardware compartilhadas, pois foi alocada uma máquina virtual para ser o servidor:

a) processador: Intel Xeon X7550;

b) quantidade de núcleos: 2;

c) memória RAM: 2GB;

[2] OpenVPN <https://openvpn.net> é uma VPN SSL open source.

d) disco: 60GB;

e) capacidade de rede: 1Gbps;

A configuração do servidor foi feita inicialmente aplicando uma camada básica de segurança, seguindo o princípio do privilégio mínimo, deixando somente as portas necessárias abertas e restringindo o acesso SSH somente ao usuário administrador. Foi elaborado um manual desse procedimento inicial. O manual está disponível para consulta no Apêndice B.

Após aplicado o princípio do privilégio mínimo ao servidor base onde será implantada a VPN, foi feita toda a instalação e configuração do serviço OpenVPN, também foi elaborado manual de todos as configurações efetuadas. O manual completo da implantação do servidor OpenVPN está disponível no Apêndice C.

4.3.1 Configuração do serviço OpenVPN

A configuração do servidor OpenVPN consiste em criar uma infraestrutura de chave privada (PKI) utilizando a ferramenta EasyRSA[3], responsável pela autoridade dos certificados SSL utilizados na VPN. É recomendado manter os servidores de PKI e de VPN em máquinas distintas (serviço de autenticação e de conexão separados), entretanto, para a finalidade deste trabalho, foi mantido um servidor monolítico.

Com a PKI operacional, são feitas as configurações do OpenVPN propriamente ditas, sendo ajustados os parâmetros para funcionamento básico do serviço conforme desejado: permitir que usuários autentiquem-se via internet e acessem a intranet através do túnel criado por essa conexão.

O sistema 2FA utiliza o certificado SSL (mantido pela PKI) e autenticação por usuário e senha (suportada pelo *plugin* PAM do Linux). A vantagem em utilizar PAM é a simplicidade de manutenção das contas de usuários, que é feita utilizando `adduser` para criar um novo usuário, `passwd` para alteração da senha e `deluser` para remoção do usuário do sistema.

Além disso, foi criado um *script* feito em Python para facilitar a criação de chaves e de usuários, ao invés de executar manualmente os comandos do EasyRSA todas as vezes que fossem criados novos usuários. O programa não consiste em uma aplicação CLI totalmente funcional, apenas é um *script* para facilitar a chamada dos comandos de maneira automática, sendo que, em caso de problemas, é demandada experiência em terminal de comando Linux para

[3] EasyRSA <https://github.com/OpenVPN/easy-rsa> é um utilitário de CA (*Certification Authority*).

lidar com erros no EasyRSA, nas contas de usuários ou no gerenciador de serviços `systemd`. É possível construir uma interface mais robusta para gestão da VPN, uma interface web por exemplo, porém está fora do escopo deste trabalho.

Além do servidor OpenVPN de produção, também é mantido um servidor OpenVPN de homologação, que está instalado em outra VPS com PKI distinta, com a finalidade de efetuar testes de funcionalidades sem afetar o ambiente em produção.

O ambiente em produção mantém conexão de dezenas de usuários simultaneamente sem perda de desempenho, servindo de porta de entrada à rede privada do provedor. A configuração atual permite que somente seja possível acessar equipamentos de núcleo da rede através da VPN, restringindo e controlando o acesso remoto a equipamentos críticos somente a usuários autorizados, minimizando a probabilidade de incidência de ataques de invasão nos mesmos.

O servidor OpenVPN principal está na rotina de backup automático dos servidores do datacenter da Minasnet, isso garante tolerância a falhas caso o servidor em produção seja corrompido. Em caso de corrompimento, é necessário somente a restauração de uma imagem do servidor para que não seja perdido nenhum certificado da PKI, deste que o *check-point* esteja em um instante após a inserção do um último usuário. Como a frequência do backup é semanal e não são adicionados novos usuários com tanta frequência, os impactos após uma inconsistência no servidor tendem a ser mínimos.

4.3.2 Considerações sobre o servidor OpenVPN

A manutenção de um servidor OpenVPN demanda perícia em ambiente Linux, pois a resolução de falhas requer análise de log do `systemd`. Um detalhe é que o serviço da VPN não se inicia automaticamente quando o servidor é reiniciado, devido à necessidade de inserção manual do *passphrase* da PKI em prompt no terminal de comandos.

Outro detalhe é que os certificados SSL têm prazo de validade de 3 anos, sendo que não é possível renová-los para estender a data de validade. É importante monitorar a validade dos certificados e gerar novos depois de vencidos para que não haja transtorno para os usuários da VPN.

4.4 Implementação de firewall

Com a VPN operacional, o próximo passo para implementação de segurança na rede é a configuração do sistema de firewall para filtragem de pacotes suspeitos. A finalidade do firewall

é mitigar as vulnerabilidades elencadas pelas notificações do CERT.br, bem como garantir a aplicação das políticas de segurança de rede adotadas pelo ISP.

A abordagem de firewall deste trabalho é simples e eficiente, fazendo filtragem por endereços IP e por números de porta, ou seja, o firewall vai inspecionar os cabeçalhos dos pacotes nas camadas de rede e de transporte. Existem modelos de firewall que inspecionam pacotes na camada de aplicação (firewall *layer* 7), porém requerem muito poder de processamento e podem afetar negativamente o *throughput* da rede, sendo dedicados às redes corporativas (universidades e grandes corporações) ao invés de provedores de internet. Filtrar IP e porta é o suficiente para a função do ISP.

Antes da implantação feita neste trabalho, não existia firewall na rede de acesso dos clientes da Minasnet, ou seja, todos os CPEs dos clientes estavam expostos à internet sem nenhum filtro, sendo a única camada de segurança a manutenção de senhas fortes nos equipamentos e configuração correta por parte dos instaladores, algo que nem sempre ocorria. Houve relatos, por exemplo, de antenas Ubiquiti e MikroTik que foram infectadas por *worms* devido às vulnerabilidades nas quais os equipamentos estavam expostos.

Após elencadas as vulnerabilidades às quais os equipamentos da rede são suscetíveis, a tarefa é implementar as regras de firewall para colocá-las em produção em um roteador que faça interface entre os tráfegos de rede, ou seja, que fique intermediando o caminho por onde passarão todos os pacotes. A abordagem deste trabalho utiliza o firewall nativo da plataforma RouterOS dos roteadores MikroTik, que possui seu funcionamento básico semelhante ao iptables do Linux, pois a próprioa plataforma é construída sobre o *kernel* Linux.

Os requisitos de firewall desenvolvidos para a Minasnet, inicialmente, são os seguintes:

a) negar o acesso remoto aos equipamentos primários dos clientes, que desempenham a função de cliente PPPoE;

b) permitir somente que o NOC e o Help Desk acessem remotamente os equipamentos dos clientes;

c) permitir que clientes de IP fixo fiquem expostos à internet, sem filtragem por firewall;

d) negar que clientes acessem a intranet do ISP;

e) aplicar filtros que corrijam vulnerabilidades detectadas na rede.

4.4.1 Regras de firewall no RouterOS

O objetivo deste firewall é filtrar conexões de entrada que caracterizem acesso ilegal a equipamentos, conforme regras definidas por IP e porta. Filtrar consiste em descartar pacotes de acordo com as regras.

A topologia utilizada para implantação do firewall aplica as regras diretamente ao concentrador (B-RAS), pois é nele que está a origem (ou destino, dependendo do ponto de vista) do tráfego dos clientes. Colocar outro equipamento intermediando o concentrador com finalidade de firewall não é suficiente, pois não é possível filtrar conexões entre clientes de um mesmo concentrador de acordo com a metodologia utilizada neste trabalho, isso porque a comutação de pacotes é feita em memória neste caso.

De acordo com a documentação do RouterOS (MIKROTIK. . . , 2020b), para implementar o firewall de acordo com os requisitos supracitados, são utilizados os seguintes parâmetros oferecidos na tabela *filter* do firewall:

a) `address-list`: estrutura de dados do tipo lista contendo endereços que serão alvo das regras, para auxiliar na construção fracamente acoplada dos componentes do firewall;

b) `action`: ação que o firewall deve executar quando a regra for acionada, como o objetivo é filtrar, será utilizada a ação *drop*;

c) `chain`: caminho que o pacote faz no roteador, como os pacotes que serão filtrados estão sendo roteados, utiliza-se a cadeia *forward*;

d) `protocol`: protocolo de transporte, TCP ou UDP;

e) `in-interface` ou `out-interface`: interfaces alvos das regras, neste caso, todos os clientes PPPoE;

f) `src-port` ou `dst-port`: número das portas que serão filtradas pelo firewall.

A construção das regras de firewall foram feitas através de *script* nativo para a plataforma. A seguir, estão exemplos da implementação de algumas regras básicas que atendem aos requisitos definidos anteriormente, sendo que, na Figura 4.4, está codificada a primeira etapa da construção do firewall: a definição das listas de IP nas quais as regras de filtragem serão trabalhadas, sendo elas as redes de gerência, que têm acesso privilegiado em toda a rede do ISP, a rede de clientes que contrataram o serviço de IP fixo e desejam ficar expostos à internet, e a

rede de Bogons, que neste caso são endereços privados do ISP e não devem ser acessíveis pelos assinantes.

Figura 4.4 – Exemplo de criação de address-list para o firewall do RouterOS.

```
/ip firewall address-list
add comment="IP FIXO CLIENTES"                    \
    address=203.0.113.128/25 list=rede_ip_fixo
add comment="SERVIDOR VPN"                        \
    address=203.0.113.123 list=rede_gerencia
add comment="HELP DESK"                           \
    address=203.0.113.124 list=rede_gerencia
add comment="ESCRITORIO NOC"                      \
    address=203.0.113.125 list=rede_gerencia
add comment="BOGONS"                              \
    address=10.0.0.0/8 list=rede_privada
add comment="BOGONS"                              \
    address=172.16.0.0/12 list=rede_privada
add comment="BOGONS"                              \
    address=192.168.0.0/16 list=rede_privada
```

Fonte: do autor (2020).

A utilização de listas de IP não é obrigatória para configuração das regras de firewall, sendo possível inserir cada um dos IPs diretamente. O problema é que a manutenibilidade do firewall fica prejudicada, assim as listas deixam os componentes fracamente conectados e permite que qualquer alteração de IP não necessite de alteração nas regras propriamente ditas, além de deixar o código mais legível, sendo assim uma boa prática na configuração do firewall.

Um dos primeiros requisitos citados para implementação do firewall foi a necessidade de impedir a invasão aos equipamentos de clientes, permitindo o acesso somente às redes gerenciais da empresa (NOC e Help Desk). O acesso remoto a esses dispositivos é feito através de página web (na maioria dos casos), como também via SSH, Telnet ou Winbox. Assim, na Figura 4.5, é apresentada a regra que determina o bloqueio de todo acesso externo nas portas de gerência em todos os assinantes, exceto IP fixo, a fim de garantir o cumprimento do requisito de segurança.

Outro requisito está codificado na Figura 4.6, uma regra para impedir a descoberta e o acesso às redes bogons, que são redes privadas. Por serem utilizados IPs privados para endereçar roteadores e servidores do núcleo da rede, é necessário bloquear toda conexão de saída em todos os clientes com destino para a lista de redes privadas (bogons), a fim de garantir o princípio do privilégio mínimo a esses equipamentos. Como se tratam de dispositivos que

Figura 4.5 – Regra de firewall para controle de acesso aos CPEs.

```
/ip firewall filter
add comment="DROP GERENCIA DE CPE"  \
    action=drop                     \
    chain=forward                   \
    out-interface=all-ppp           \
    src-address-list=!rede_gerencia \
    dst-address-list=!rede_ip_fixo  \
    dst-port=0-1023                 \
    protocol=tcp
```

Fonte: do autor (2020).

operam a parte crítica da rede, devem ser acessíveis somente à equipe do NOC, pois se algum hacker conseguir elencar e explorar alguma vulnerabilidade nesses equipamentos, pode causar um prejuízo enorme ao ISP.

Figura 4.6 – Regra de firewall para bloqueio de acesso à rede privada do provedor.

```
/ip firewall filter
add comment="DROP REDE BOGON"     \
    action=drop                   \
    chain=forward                 \
    in-interface=all-ppp          \
    dst-address-list=rede_privada
```

Fonte: do autor (2020).

A Figura 4.7 tem a implementação do filtro que impede ataque de *spam* a partir de servidores SOCKS abertos em Mikrotik, bloqueando toda conexão de entrada na porta 4145 TCP em todos os assinantes, sem exceção, por razões de segurança, uma vez que a notificação do CERT.br disponível no Anexo B relata que inclusive clientes de IP fixo estão vulneráveis. Como essa regra foi desenvolvida com a finalidade de corrigir o problema citado no e-mail, todos os assinantes são alvo dessa filtragem.

Não será exposto o script completo de configuração do firewall dos concentradores da Minasnet neste documento, por questão de segurança e de confidencialidade, uma vez que pessoas mal intencionadas poderiam explorar alguma vulnerabilidade que tenha passada despercebida na metodologia utilizada. Mesmo que não exista sistema 100% seguro, mantendo confidencialidade é possível dificultar o trabalho dos atacantes. Por isso, ficaram descritos aqui somente esses exemplos.

Figura 4.7 – Regra de firewall para correção da vulnerabilidade por SOCKS notificada pelo CERT.br.

```
/ip firewall filter
add comment="DROP SOCKS 4145" \
    action=drop               \
    chain=forward             \
    out-interface=all-ppp     \
    dst-port=4145             \
    protocol=tcp
```

Fonte: do autor (2020).

4.4.2 Considerações sobre o firewall

A criação de regras de firewall é uma tarefa simples quando se conhece os recursos da plataforma na qual está trabalhando, porém requer cautela, pois configuração errônea pode comprometer a navegação dos assinantes ou então afetar negativamente o processamento dos roteadores nos quais foram feitos a implantação das regras. Devido à familiaridade com iptables, o firewall do RouterOS é intuitivo para quem tem experiência com o sistem Linux, entretanto é um recurso implantado em software. O fato de o firewall não possuir hardware dedicado para a tarefa, limita a capacidade de filtragem que pode ser aplicada através dele.

Após colocar o firewall da Minasnet em produção, notificações recorrentes do CERT.br cessaram. No entanto, novas vulnerabilidades são elencadas frequentemente pelos *scanners* do grupo. Esses eventos determinam a necessidade da contínua tarefa de criação, de melhoria e de aperfeiçoamento das regras do firewall dos concentradores para atender aos novos requisitos de segurança levantados.

A implantação de firewall no concentrador requer cuidado, uma vez que pode comprometer o desempenho da Routerboard. Por isso, foi feito monitoramento de consumo de CPU em diferentes horários do dia, inclusive em horários de pico por volta das 20h, para constatar que o firewall desenvolvido não afetou o desempenho do roteador consideravelmente, sendo que o processamento do concentrador já era concorrido pela manutenção das *queues* e do CGNAT.

A dificuldade em implantar regras de firewall é o risco de afetar a navegação no ambiente em produção, sendo necessário ambiente de testes antes de ser feito o *deploy*. Na Minasnet, foram utilizados conjuntos de RB750 e pequenos servidores web instalados no computador para testar a filtragem dos pacotes antes de colocar em operação. Também é possível fazer os testes através de simuladores, como o GNS3, no qual é possível construir um ambiente de homologação completo para redes e sistemas distribuídos.

5 CONCLUSÃO

No curso de Bacharelado em Ciência da Computação da Universidade Federal de Lavras, o estudante tem a oportunidade de aprender os aspectos teóricos que fundamentam as vastas tecnologias digitais que são indispensáveis à sociedade contemporânea, as Tecnologias da Informação e Comunicação (TICs). Dentro das TICs, destaca-se uma das maiores e mais poderosas ferramentas desenvolvida pelo ser humano: a internet, que foi o destaque neste trabalho.

O trabalho serviu como ponte entre o conhecimento acadêmico e o conhecimento prático. Cálculos de sub-redes podiam até parecer não ter sentido em listas de exercícios em disciplina da graduação, mas com os cálculos de endereçamento e de implantação de CGNAT mostraram sua importância para garantir que a camada lógica da internet funcione e permita que os assinantes naveguem pela rede. Não somente isso, adotar práticas e políticas de segurança com adoção de firewall e VPN também possibilitou a aplicação dos princípios que antes foram abordados somente em laboratório e em estudos de caso fictícios.

O legado gerencial deixado pelo estágio está no monitoramento contínuo dos serviços como estratégia de inteligência de negócios na empresa, pois é através dele que é possível reagir às falhas o quanto antes, bem como prever problemas, a fim de garantir a qualidade do serviços prestados aos clientes. O trabalho abordou a parte mais técnica de operação de redes, mas o dia a dia gira em torno da gestão e do monitoramento de incidentes.

Os objetivos foram alcançados sem muitas dificuldades operacionais para a execução dos mesmos, pois o estagiário já tinha uma bagagem teórica consolidada e o trabalho serviu como meio para praticar o conhecimento e gerar valor. As dificuldades foram recorrentes devido à insegurança em aplicar alterações em sistemas em produção com milhares de clientes acessando, algo normal quando se pensa que um erro pode "parar tudo" em questão de segundos.

Apesar de na universidade existirem atividades e projetos em que o estudante pode obter conhecimento prático dos tópicos abordados em sala de aula, através de trabalhos práticos em laboratórios ou simuladores, de projetos de extensão acadêmica ou mesmo em empresas juniores, o estudante não conseguirá uma visão ampla da dimensão do assunto, por estar lidando com casos particulares e restritos. Por isso, é importante a realização do estágio em uma organização, onde se pode aprender e desenvolver habilidades práticas em um ambiente em produção e que presta serviço a toda a comunidade, seja em nível local, regional, nacional ou internacional.

Por isso a realização deste trabalho de estágio supervisionado na organização Minasnet foi importante para a maturação do estudante como profissional da área de redes, pois foi o momento quando se colocou em prática os protocolos aprendidos na disciplina de Redes de Computadores como também as arquiteturas de sistemas distribuídos, em nível de infraestrutura, vistos na disciplina de Arquitetura de Computadores e em Sistemas Distribuídos. De fato, o estágio abordou assuntos que foram além dessas disciplinas básicas da área de infraestrutura, o que demonstra uma falta de disciplinas ofertadas nessa área por parte do Departamento de Ciência da Computação. Durante o curso, foram mínimas (se não inexistentes) disciplinas eletivas sobre assuntos de redes, o que está deixando o curso de Ciência da Computação focado quase que exclusivamente na camada de aplicação do Modelo OSI, com foco em desenvolvimento de aplicações. Não se pode deixar de lado a infraestrutura dos sistemas de computação e de informação, pois é ela que sustenta todas as aplicações que estão em operação, por exemplo, na web.

REFERÊNCIAS

BRASIL. **Lei nº 12.965, de 23 de abril de 2014**: Estabelece princípios, garantias, direitos e deveres para o uso da internet no brasil. Brasília: Imprensa Nacional, 2014. Lei nº 12.965/2014. (Diário Oficial da União, 77). Disponível em: <http://pesquisa.in.gov.br/imprensa/jsp/visualiza/index.jsp?data=24/04/2014&jornal=1&pagina=1&totalArquivos=124>. Acesso em: 23 mai. 2020.

CERT. **Resultados Preliminares do Projeto SpamPots: Uso de Honeypots de Baixa Interatividade na Obtenção de Métricas sobre o Abuso de Redes de Banda Larga para o Envio de Spam**. CERT.br, 2007. Disponível em: <https://cert.br/docs/whitepapers/spampots/>. Acesso em: 21 jun. 2020.

IANA. **Number Resources**. Internet Assigned Numbers Authority, 2020. Disponível em: <https://www.iana.org/numbers>. Acesso em: 19 jul. 2020.

KEPIOS. **Digital 2020 Brazil**. DateReportal, 2020. Disponível em: <https://datareportal.com/reports/digital-2020-brazil>. Acesso em: 5 jul. 2020.

KUROSE, J. F.; ROSS, K. W. **Redes de computadores e a Internet**: uma abordagem top-down. 6. ed. São Paulo: Pearson, 2014. 634 p.

LEVINE, J. R. **DNS Blacklists and Whitelists**. RFC Editor, 2010. RFC 5782. (Request for Comments, 5782). Disponível em: <https://rfc-editor.org/rfc/rfc5782.txt>. Acesso em: 14 jun. 2020.

MAIA, W. From IPv4 Scarcity to IPv6 Abundance. In: EUROPEAN MUM 2018. Berlin: MikroTik User Meeting, 2018. Disponível em: <https://mum.mikrotik.com/presentations/EU18/presentation_5195_1524667160.pdf>. Acesso em: 28 mai. 2020.

MIKROTIK documentation: Manual:IP/Firewall/NAT. MikroTik wiki, 2020. Disponível em: <https://wiki.mikrotik.com/wiki/Manual:IP/Firewall/NAT>. Acesso em: 24 mai. 2020.

MIKROTIK documentation: Manual:IP/Firewall/Filter. MikroTik wiki, 2020. Disponível em: <https://wiki.mikrotik.com/wiki/Manual:IP/Firewall/Filter>. Acesso em: 25 jun. 2020.

MOSKOWITZ, R. et al. **Address Allocation for Private Internets**. RFC Editor, 1996. RFC 1918. (Request for Comments, 1918). Disponível em: <https://rfc-editor.org/rfc/rfc1918.txt>. Acesso em: 22 mai. 2020.

NAKAMURA, E. T.; GEUS, P. L. de. **Segurança de redes em ambientes cooperativos**. 2. ed. São Paulo: Novatec Editora, 2007. 482 p.

PERREAULT, S. et al. **Common Requirements for Carrier-Grade NATs (CGNs)**. RFC Editor, 2013. RFC 6888. (Request for Comments, 6888). Disponível em: <https://rfc-editor.org/rfc/rfc6888.txt>. Acesso em: 23 mai. 2020.

WEIL, J. et al. **IANA-Reserved IPv4 Prefix for Shared Address Space**. RFC Editor, 2012. RFC 6598. (Request for Comments, 6598). Disponível em: <https://rfc-editor.org/rfc/rfc6598.txt>. Acesso em: 16 mai. 2020.

APÊNDICE A – Documentação de leitura rápida do py-cgnat

py-cgnat

Python module for generating CGNAT rules using netmap

Brief

Python library and CLI program for generating firewall rules to deploy Carrier-Grade NAT, besides translating a given IP and port to its private address and vice versa. The methodology consists in building netmap rules at 1:32 public-private ratio, mapping a range of 2.000 ports for each client. Works for any netmask, since that follow the 1:32 ratio:

Private prefix	Public prefix	N. of clients
...	...	...
/20	/25	4096
/21	/26	2048
/22	/27	1024
/23	/28	512
/24	/29	256
/25	/30	128
/26	/31	64
/27	/32	32

Supported Platforms

- MikroTik RouterOS

Requirements

- Python 3.7+

How to install it?

Installation can just being done with `pip`:

```
pip install pycgnat
```

How to use it?

1. Command Line Interface

For **generating** the rules, you can print it in console or save it to a file:

```
pycgnat 100.64.0.0/20 203.0.113.0/25 gen routeros filename.rsc
pycgnat 100.64.0.0/20 203.0.113.0/25 gen routeros
```

For **translating** a private IP to its public one, use the `direct` option:

```
pycgnat 100.64.0.0/20 203.0.113.0/25 trans --direct 100.64.2.15
pycgnat 100.64.0.0/20 203.0.113.0/25 trans -d 100.64.2.15
```

For **translatig** a public IP and port to its private IP correspondent, use the `reverse` option:

```
pycgnat 100.64.0.0/20 203.0.113.0/25 trans --reverse 203.0.113.20:13578
pycgnat 100.64.0.0/20 203.0.113.0/25 trans -r 203.0.113.20:13578
```

The CLI includes useful **help** command (supported by `argparse` framework), so just type:

```
pycgnat --help
pycgnat -h
```

2. Python library

You can use the functionalities directly in Python lang. Just **import** the wanted module to your program:

```
from pycgnat.translator.reverse import cgnat_reverse

dic = cgnat_reverse(privnet, pubnet, IPv4Address('203.0.113.20'), 13578)
print(dic['private_ip'])
```

The full `pycgnat`'s documentation is written in the source-code.

Future works

- Add support for other platfoms (I'm using MikroTik for while, so this is the reason for only supporting it at first version).

APÊNDICE B – Manual para configuração básica de segurança em servidor Debian 9

O objetivo deste tutorial é criar uma camada básica de segurança em servidor.

Índice

1. Criando um Usuário Administrador
2. Criando um Firewall Básico
3. Restringindo o Acesso SSH

Passo 1:

Criando um Usuário Administrador

> O procedimento inicial deve ser feito através do usuário `root`.

Primeiramente, deve ser instalado o comando `sudo`.

```
# apt install sudo
```

Após a instalação, criar um usuário não `root` com privilégios administrativos no sistema.

```
# adduser administrador
```

Para dar privilégios administrativos ao usuário criado, adicione-o ao grupo `sudo`:

```
# usermod -aG sudo administrador
```

Com o usuário criado e com privilégios no sistema, devemos usá-lo daqui em diante.

```
# su - administrador
```

Passo 2:

Criando um Firewall Básico

Instale o pacote `ufw`.

```
$ sudo apt install ufw
```

Habilite a conexão SSH no recém instalado firewall.

```
$ sudo ufw allow OpenSSH
```

Feito isso, habilite o deamon do firewall.

```
$ sudo ufw enable
```

Neste ponto, após habilitar o firewall, **toda conexão no servidor está sendo bloqueada, exceto SSH** devido à configuração anterior. Sempre que for colocar um novo serviço em execução que precise de conexão remota, adicione a devidade regra no firewall `ufw`.

Passo 3:

Restringindo o Acesso SSH

O acesso SSH deve ser permitido somente ao administrador. Por isso, abra as configurações de `sshd`.

```
$ sudo nano /etc/ssh/sshd_config
```

Adicione a seguinte linha no final do arquivo:

```
AllowUsers administrador
```

Salve e feche o arquivo. Então reinicie o serviço SSH.

```
$ sudo systemctl restart sshd
```

Agora somente o usuário administrador pode acessar ao servidor remotamente. Se for preciso liberar o acesso a outro usuário, deve ser adicionado o `username` do mesmo em `AllowUsers`, separado somente por espaços entre os nomes.

> PRONTO!

Referências

- https://www.digitalocean.com/community/tutorials/initial-server-setup-with-debian-9
- https://www.ostechnix.com/allow-deny-ssh-access-particular-user-group-linux/

APÊNDICE C – Manual para instalação de servidor OpenVPN no Debian 9

O objetivo deste tutorial é fazer o deploy de uma VPN usando OpenVPN.

Índice

1. Instalando OpenVPN e EasyRSA
2. Configurando as Variáveis do EasyRSA e Construindo o CA
3. Criando o Server Certificate, Key e Encryption Files
4. Configurando o Serviço OpenVPN
5. Ajustando as Configurações de Rede do Servidor
6. Iniciando e Habilitando o Serviço OpenVPN
7. Configurando Camanda de Autenticação por Usuário e Senha
8. Criando a Infraestrutura de Configuração de Clientes

Passo 1:

Instalando OpenVPN e EasyRSA

De início, instale o OpenVPN.

```
$ sudo apt update
$ sudo apt install openvpn
```

OpenVPN é uma VPN TLS/SSL. Isso significa que utiliza certificados para criptografar o tráfego na rede. Para configuração de certificados confiáveis, deve ser criado seu próprio certificate authority (CA). Para isso, é necessário a versão mais recente do EasyRSA, que será usado para construir a CA public key infrastructure (PKI).

Então, faça o download da release do EasyRSA. Para isso, localize o link de através da página https://github.com/OpenVPN/easy-rsa/releases/latest, copie-o e faça o download usando o `wget` como no exemplo a seguir:

```
$ wget -P ~/ https://github.com/OpenVPN/easy-rsa/releases/download/v3.0.6/EasyRSA-unix-v3.0.6.tgz
```

Então extraia o tarball no diretório `home`:

```
$ cd ~
$ tar xvf EasyRSA-unix-v3.0.6.tgz
```

Renomeie o diretório do arquivo extraído para somente EasyRSA:

```
$ mv EasyRSA-v3.0.6/ EasyRSA/
```

Exclua o arquivo baixado, ele não é mais necessário.

```
$ rm EasyRSA-unix-v3.0.6.tgz
```

Assim está finalizada a instalação da infraestrutura básica da VPN.

Passo 2:

Configurando as Variáveis do EasyRSA e Construindo o CA

Vá para o diretório do EasyRSA:

```
$ cd ~/EasyRSA/
```

Dentro do directory existe um arquivo nomeado `vars.example`. Faça a cópia dele and renomeie-a para `vars`, sem extensão de arquivo:

```
$ cp vars.example vars
```

Abra o arquivo `vars`:

```
$ nano vars
```

Encontre as configurações que definem campos padrão para novos certificados. Tem a seguinte aparência no arquivo:

[~/EasyRSA/vars]

```
. . .
#set_var EASYRSA_REQ_COUNTRY    "US"
#set_var EASYRSA_REQ_PROVINCE   "California"
#set_var EASYRSA_REQ_CITY       "San Francisco"
#set_var EASYRSA_REQ_ORG        "Copyleft Certificate Co"
#set_var EASYRSA_REQ_EMAIL      "me@example.net"
#set_var EASYRSA_REQ_OU         "My Organizational Unit"
. . .
```

Descomente essas linhas e altere os valores padrão para os seguintes:

[~/EasyRSA/vars]

```
. . .
set_var EASYRSA_REQ_COUNTRY     "BR"
set_var EASYRSA_REQ_PROVINCE    "Minas Gerais"
set_var EASYRSA_REQ_CITY        "Perdoes"
set_var EASYRSA_REQ_ORG         "Minasnet Servicos de Provedor de Internet Ltda"
set_var EASYRSA_REQ_EMAIL       "atendimento@minasnet.net"
set_var EASYRSA_REQ_OU          "Centro de Gerenciamento de Redes"
. . .
```

Ao final, salve e feche o arquivo.

Dentro do diretório EasyRSA tem um script chamado `easyrsa` que é usado para executar uma variedade de tarefas envolvendo criação e manutenção do CA. Execute o script com a opção `init-pki` para iniciar a public key infrastructure:

```
$ ./easyrsa init-pki
```

Entre com os seguintes parâmetros dentro do prompt:

- **Passphrase:** ********
- **Common Name:** SERVER-OPENVPN

> Mantenha senhas secretas!

Após isso, execute:

```
$ ./easyrsa build-ca
```

Isso contruirá o CA e criará dois arquivos importantes — `ca.crt` e `ca.key` — que compõem os lados público e privado do certificado SSL.

Com isso, o CA está pronto para assinar requisições certificadas.

Passo 3:

Criando o Server Certificate, Key e Encryption Files

Execute `easyrsa`, desta vez usando a opção `gen-req` seguida do common name da máquina.

```
$ ./easyrsa gen-req SERVER-OPENVPN
```

Isso cria a private key do servidor no arquivo `server.req`. Copie o server key para `/etc/openvpn/`:

```
$ sudo cp ~/EasyRSA/pki/private/SERVER-OPENVPN.key /etc/openvpn/
```

Então assine a requição executando `easyrsa` com a opção `sign-req`, seguida pelo *request type* e o *common name*:

```
$ ./easyrsa sign-req server SERVER-OPENVPN
```

Depois, copie os arquivos `server.crt` e `ca.crt` para `/etc/openvpn/`:

```
$ sudo cp pki/issued/SERVER-OPENVPN.crt /etc/openvpn/
$ sudo cp pki/ca.crt /etc/openvpn/
```

Crie uma chave forte de Diffie-Hellman para ser usada durante a troca de chaves:

```
$ ./easyrsa gen-dh
```

Isso pode demorar alguns minutos. Após o término, gere uma assinatura HMAC para fortalecer a verificação de integridade TLS do servidor:

```
$ sudo openvpn --genkey --secret ta.key
```

Quando terminar, copie os dois novos arquivos para `/etc/openvpn/`:

```
$ sudo cp ta.key /etc/openvpn/
$ sudo cp pki/dh.pem /etc/openvpn/
```

Com isso, todos os arquivos de chave de certificação necessários ao servidor forma gerados. A infraestrutura está pronta para gerar as chaves correspondentes para os clientes que irão acessar ao servidor OpenVPN.

Passo 4:

Configurando o Serviço OpenVPN

Inicie copiando o arquivo de exemplo de configuração OpenVPN e extraia-o para servir de base para o servidor:

```
$ sudo cp /usr/share/doc/openvpn/examples/sample-config-files/server.conf.gz /etc/openvpn/
$ sudo gzip -d /etc/openvpn/server.conf.gz
```

Abra o arquivo de configuração extraído:

```
$ sudo nano /etc/openvpn/server.conf
```

Encontre a seção HMAC pesquisando pela diretiva `tls-auth`. Esta linha deve estar descomentada, então remova o ";" para descomentá-la. Abaixo desta linha, adicione o parâmetro `key-direction`, com valor "0":

[/etc/openvpn/server.conf]

```
tls-auth ta.key 0 # This file is secret
key-direction 0
```

Depois, encontre a seção de cryptographic ciphers pesquisando pelas linhas comentadas `cipher`. A cifra `AES-256-CBC` oferece bom nível de criptografia e é bem suportada. Novamente, esta linha deve ser descomentada, remova o ";" se houver. Abaixo dela, adicione a diretiva `auth` para selecionar o algoritmo HMAC message digest, sendo `SHA256` uma boa escolha:

[/etc/openvpn/server.conf]

```
cipher AES-256-CBC
auth SHA256
```

Depois, encontre a linha contendo a diretiva `dh` que define os parâmetros de Diffie-Hellman. Se necessário, mude o nome de arquivo listado aqui removendo o `2048` de forma que fique com o nome do arquivo gerado anteriormente:

[/etc/openvpn/server.conf]

```
dh dh.pem
```

Encontre as configurações `user` e `group` e remova o ";" no início da linha:

[/etc/openvpn/server.conf]

```
user nobody
group nogroup
```

Modifique as linhas de `cert` e `key` para os apropriados arquivos `.crt` e `.key`, devendo ficar desta forma:

[/etc/openvpn/server.conf]

```
cert SERVER-OPENVPN.crt
key SERVER-OPENVPN.key
```

Também altere o protocolo para `tcp`, descomentando `tcp` e comentando `udp`:

[/etc/openvpn/server.conf]

```
proto tcp;
;proto udp;
```

Após alterar para TCP, será necessário mudar o valor da diretiva `explicit-exit-notify` de 1 para 0, pois ela é utilizada apenas pelo UDP. Não mudar esse valor pode causar erro no serviço OpenVPN:

[/etc/openvpn/server.conf]

```
explicit-exit-notify 0
```

As configurações feitas até aqui não força o tráfego dos dados pela VPN após a conexão do cliente. Para forçar os clientes a usarem o túnel da VPN, é necessário proceder com as configurações a seguir.

Encontre a seção `redirect-gateway` e remova o ";" no início da linha para deixá-la descomentada:

[/etc/openvpn/server.conf]

```
push "redirect-gateway def1 bypass-dhcp"
```

Abaixo dela, encontre a seção `dhcp-option`. Novamente, remova o ";" e deixe da seguinte forma:

[/etc/openvpn/server.conf]

```
push "dhcp-option DNS 177.66.48.12"
push "dhcp-option DNS 177.66.48.13"
```

As mudanças feitas no arquivo de exemplo `server.conf` até aqui são necessárias para o pleno funcionamento do servidor e da conexão.

Passo 5:

Ajustando as Configurações de Rede do Servidor

Ajuste a configuração padrão de redirecionamento de IP do servidor modificando o arquivo `/etc/sysctl.conf`:

```
$ sudo nano /etc/sysctl.conf
```

Dentro, procure pela linha comentada que coloca `net.ipv4.ip_forward`. Remova o "#" do início da linha para descomentar essa configuração:

[/etc/sysctl.conf]

```
net.ipv4.ip_forward=1
```

Salve e feche o arquivo.

Para carregar as novas configurações para a sessão atual, digite:

```
$ sudo sysctl -p
```

Algumas das configurações do firewall devem ser modificadas para habilitar o mascaramento. Antes de abrir as configurações do firewall e habilitar o mascaramento, primeiramente deveser encontrada a interface de rede pública do servidor. Para isso, digite:

```
$ ip route | grep default
```

Com a interface associada à sua rota padrão, abra o arquivo `/etc/ufw/before.rules` para adicionar a configuração necessária:

```
$ sudo nano /etc/ufw/before.rules
```

No início do arquivo, adicione as linhas abaixo:

[/etc/ufw/before.rules]

```
#
# rules.before
#
# Rules that should be run before the ufw command line added rules. Custom
# rules should be added to one of these chains:
#   ufw-before-input
#   ufw-before-output
#   ufw-before-forward
#

# START OPENVPN RULES
# NAT table rules
*nat
:POSTROUTING ACCEPT [0:0]
# Allow traffic from OpenVPN client to eth0 (change to the interface you discovered!)
-A POSTROUTING -s 10.8.0.0/8 -o eth0 -j MASQUERADE
COMMIT
# END OPENVPN RULES

# Don't delete these required lines, otherwise there will be errors
*filter
. . .
```

Salve e feche o arquivo quando concluir.

Depois, voce tem que configurar o UFW para permitir redirecionamento de pacotes por padrão. Para isso, abra o arquivo `/etc/default/ufw`:

```
$ sudo nano /etc/default/ufw
```

Dentro, encontre a diretiva `DEFAULT_FORWARD_POLICY` e modifique o valor de `DROP` para `ACCEPT`:

[/etc/default/ufw]

```
DEFAULT_FORWARD_POLICY="ACCEPT"
```

Salve e feche o arquivo.

Agora, ajuste o firewall para habilitar o tráfego do OpenVPN:

```
$ sudo ufw allow 1194/tcp
```

Após adicionar a regra no firewall, desabilite e reabilite o UFW para reiniciá-lo:

```
$ sudo ufw disable
$ sudo ufw enable
```

O servidor está agora pronto para lidar com tráfego da VPN.

Passo 6:

Iniciando e Habilitando o Serviço OpenVPN

O servidor está pronto para inciar o serviço OpenVPN. Antes de iniciá-lo, o passphrase é pedido pelo `systemd`, apenas digite o seguinte comando e preencha o prompt com o passphrase definido anteriormente:

```
$ sudo systemd-tty-ask-password-agent
```

Inicie o servidor OpenVPN:

```
$ sudo systemctl start openvpn@server
```

Verifique se o serviço foi iniciado com sucesso:

```
$ sudo systemctl status openvpn@server
```

Você também pode verificar se a interface do túnel OpenVPN `tun0` está disponível:

```
$ ip addr show tun0
```

O servidor OpenVPN em execução neste momento, se nenhum erro tiver ocorrido. Devido ao passphrase definido no EasyRSA, não é possível iniciar o serviço automaticamente no boot, devendo, portantos, ser iniciado manualmente com os comandos definidos nesta seção.

Passo 7:

Configurando Camanda de Autenticação por Usuário e Senha

Até este momento, o servidor OpenVPN em execução funciona pela autenticação por troca de chaves. Para aumentar o nível de segurança do servidor OpenVPN, será adicionado mais uma camada de autenticação, exigindo, além das chaves, login e senha através do Linux PAM.

Para isso, modifique o arquivo `server.conf` para habilitar o plugin do PAM:

```
$ sudo nano /etc/openvpn/server.conf
```

Adicone ao final dele a seguinte linha:

[/etc/openvpn/server.conf]

```
plugin /usr/lib/openvpn/openvpn-plugin-auth-pam.so login
```

Salve e feche o arquivo. Após, reinicie o servidor para que o novo módulo seja carregado corretamente:

```
$ sudo reboot
```

Após reiniciar, execute o script de início do servidor OpenVPN, que está no diretório do administrador, já que não é possível iniciá-lo automaticamente:

```
$ ./start-openvpn
```

Digite o passphrase e pronto. Finalmente, verifique se o servidor está em produção:

```
$ sudo systemctl status openvpn@server
```

Se o status não indicar nenhum erro, a VPN está pronta para uso.

Passo 8:

Criando a Infraestrutura de Configuração de Clientes

Primeiramente, faça o download da base de configurações de gerenciamento de clientes OpenVPN:

```
$ wget -P /tmp/ https://williamabreu.github.io/MNET-tutorials/openvpn/download/mnet-openvpn-configs.tar.gz
```

Depois, extraia o tarball:

```
$ cd /tmp/
$ tar xvf mnet-openvpn-configs.tar.gz
```

Entre no diretório extraído e execute o autoconfigurador:

```
$ cd mnet-openvpn-configs/
$ ./configure
```

Agora com a base de configurações de clientes OpenVPN instaladas em `~/client-configs/`, navegue até o diretório raíz do administrador:

```
$ cd ~
```

Neste diretório estão os scripts para facilitar a criação de usuários e início do servidor OpenVPN, são os executáveis `add-openvpn-user` e `start-openvpn`.

Para adicionar um novo usuário, como exemplo o de username sendo `client1`, execute o comando:

```
./add-openvpn-user client1
```

Preencha os dados do prompt corretamente.

Assim que terminar o processo, o arquivo que deve ser enviado para o cliente para que ele possa acessar à VPN está disponível em `~/client-configs/ovpn-files/`, sendo o arquivo cujo nome é o respectivo username com extensão `.ovpn`. Como no exemplo, o arquivo seria `client1.ovpn`. Esse arquivo deve ser enviado por um meio seguro e não pode ser compartilhado.

Repita este último procedimento sempre que for criar um novo usuário da VPN.

> PRONTO!

Referências

- https://www.digitalocean.com/community/tutorials/how-to-set-up-an-openvpn-server-on-debian-9
- https://www.youtube.com/watch?v=V6DGD4QRXVU

ANEXO A – Um relatório justificando listagem de IP na blacklist CBL

CBL LOOKUP

IMPORTANT: Many CBL/XBL listings are caused by a vulnerability in Mikrotik routers. If you have a Mikrotik router, please check out the Mikrotik blog on this subject (https://blog.mikrotik.com/security/winbox-vulnerability.html) and follow the instructions before attempting to remove your CBL listing.

IP Address: 177.93.97.228

RESULTS OF LOOKUP

177.93.97.228 is listed

This IP address was detected and listed 2 times in the past 28 days, and 0 times in the past 24 hours. The most recent detection was at Mon Mar 4 21:55:00 2019 UTC +/- 5 minutes

This IP address is infected with, or is NATting for a machine infected with the Conficker malicious botnet.

More information about Conficker can be obtained from Wikipedia (http://en.wikipedia.org/wiki/Conficker)

Please follow these instructions.

Dshield (http://www.dshield.org/diary/Third+party+information+on+conficker/5860) has a diary item containing many third party resources, especially removal tools such as Norton Power Eraser, Stinger, MSRT

etc.

One of the most critical items is to make sure that all of your computers have the MS08-067 patch installed. But even with the patch installed, machines can get reinfected.

There are several ways to identify Conficker infections remotely. For a fairly complete approach, see Sophos (http://www.sophos.com/en-us/support/knowledgebase/61259.aspx).

If you have full firewall logs turned on at the time of detection, this may be suficient to find the infection on a NAT:

This was detected by a TCP connection from "177.93.97.228" on port "40170" going to IP address "38.229.146.66" (the sinkhole (sinkhole.html)) on port "80".

The botnet command and control domain for this connection was "n/a".

This detection corresponds to a connection at Mon Mar 4 21:58:48 2019 UTC (this timestamp is believed accurate to within one second).

Detection Information Summary	
Destination IP	38.229.146.66
Destination port	80
Source IP	177.93.97.228
Source port	40170
C&C name/domain	n/a
Protocol	TCP
Time	Mon Mar 4 21:58:48 2019 UTC

Behind a NAT, you should be able to find the infected machine by looking for attempted connections to IP address "38.229.146.66" or host name "n/a" on any port with a network sniffer such as Wireshark. Equivalently, you can examine your DNS server or proxy server logs to references to "38.229.146.66" or "n/a". See Advanced Techniques (advanced.html) for more detail on how to use Wireshark - ignore the references to port 25/SMTP traffic - the identifying activity is NOT on port 25.

Please note that some of the above quoted information may be empty ("") or "na" or "-". In those cases, the feed has declined or is unable to give us that information. Hopefully enough information will be present to allow you to pinpoint the connections. If not, the destination ports to check are usually port 80, 8080, 443 or high ports (around 16000) outbound from your network. Most of these infections make very large numbers of connections; they should stand out.

If you don't have full firewall logging, perhaps you can set up a firewall block/log of all access (any port) to IP address 38.229.146.66 and keep watch for hits.

Recent versions of NMap (http://insecure.org/) can detect Conficker, but it's not 100% reliable at finding every infection. Nmap is available for Linux, xxxBSD, Windows and Mac. Nessus can also find Conficker infections remotely. Several other scanners are available here (http://net.cs.uni-bonn.de/wg/cs/applications/containing-conficker/).

Enigma Software's scanner (http://www.enigmasoftware.com/a1/download/cfremover.exe) is apparently good at finding Conficker A.

University of Bonn (http://net.cs.uni-bonn.de/wg/cs/applications/containing-conficker/) has a number of scan/removal tools.

If you're unable to find the infection, consider:

1. If you used a network scanner, make sure that the network specification you used to check your network was right, and you understand how to interpret a conficker detection.
2. Some network conficker scanners only detect some varieties of conficker. For example, nmap misses some. If you can't find it with nmap, try other scanners like McAfee's (http://www.mcafee.com/ca/threat-center/confickertest.aspx). In other words, try at least two.
3. Are you sure you have found _all_ computers in your network? Sometimes there are machines quietly sitting in back rooms somewhere that got forgotten about. It would be a good idea to run

   ```
   nmap -sP <ALL of your network specifications>
   ```

   which should list all your computers, printers and other network devices. Did you see all the computers you expected to see?
4. The infected computer may be turned off at the time you ran the scan or not on the network. Double-check everything was turned on during the scan.
5. If you have wireless, make sure it's secured with WPA or WPA2, and that "strangers" can't connect. WEP security is NOT good enough.
6. Many versions of Conficker propagate via infected thumbdrives/USB keys. When an infected machine is found, ALL such devices associated with the machine should be considered suspect, and either destroyed or completely reformatted.
7. Conficker also propagates by file and printer shares.

SELF REMOVAL:

Normally, you can remove the CBL listing yourself. If no removal link is given below, follow the instructions, and

come back and do the lookup again, and the removal link will appear.

I have verified that all of my computers and services accessible from the Internet through this IP address (computers, such as external router admin interfaces, web servers, Internet of Things devices such as DVRs, webcams and Baby Cameras) all have inwards Internet access turned off, OR, have had their passwords changed from the default factory setting.	REMOVE

ANEXO B – Um e-mail de alerta de segurança enviado pelo CERT.br

From: "CERT.br" <cert@cert.br>
Subject: Alerta: [AS 262488] Mikrotik Possivelmente Comprometido - SOCKS 4145
Date: 3 June 2019 09:54:05 GMT-3
To: abuse@minasnet.net
Cc: cert@cert.br
Reply-To: cert@cert.br

Caro responsável,

Os IPs no log ao final dessa mensagem possivelmente são de dispositivos Mikrotik em sua rede que foram comprometidos e que estão sendo abusados intensamente para o envio de spam.

Esse comprometimento habilita o serviço SOCKS na porta 4145/tcp
que pode ser abusado para diversas atividades, principalmente para o envio de spam.

Essas atividades estão consumindo recursos de sua rede e provavelmente incluindo seus IPs em listas de bloqueio.

Se você não for a pessoa correta para corrigir este problema, por favor repasse essa mensagem para alguém da sua organização que possa fazê-lo.

Gostaríamos de solicitar que:

1. cada dispositivo associado aos IPs abaixo fosse revisado e, se confirmada a suspeita de comprometimento, o problema seja resolvido (com a desativação do serviço SOCKS, alteração das senhas e atualização do RouterOS);
2. aumente-se o nível de monitoração da rede para determinar se outros dispositivos da sua rede também estão sofrendo do mesmo problema.

Sugestões de como realizar esses dois itens seguem abaixo.

* Como resolver o comprometimento dos dispositivos Mikrotik?

1. Verifique a existência de um serviço SOCKS atendendo na
   porta 4145/tcp, executando o seguinte comando:
   /ip socks print
   Se o serviço estiver marcado como habilitado (enabled = yes),
   desabilite-o com o seguinte comando:
   /ip socks set enable=no
2. Atualize a versão do Router OS para a última versão
   "Long-term/bugfix" ou "Stable/current", de acordo com as instruções
   do fabricante disponíveis na seguinte URL:
   https://wiki.mikrotik.com/wiki/Manual:Upgrading_RouterOS
3. Apenas depois de atualizar o sistema altere a senha com o comando
   abaixo:
   /user set USUARIO password=NOVA_SENHA
   onde USUARIO é o usuário utilizado para conectar no mikrotik

* Como identificar outros dispositivos sendo abusados pela mesma técnica?

Sugerimos também que monitore regularmente o tráfego de sua rede através do uso de netflow para identificar esse problema. Uma sugestão seria monitorar o tráfego na porta 4145/tcp e observar o aumento anormal de conexões com destino às portas 25/tcp e 587/tcp.

* O que é o CERT.br?

O CERT.br -- Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil -- é o Grupo de Resposta a Incidentes de Segurança para a Internet brasileira, mantido pelo NIC.br do Comitê Gestor da Internet no Brasil. É o grupo responsável por tratar incidentes de segurança em computadores, envolvendo redes conectadas

à Internet brasileira.

IP	ASN	Porta	Status	Timestamp	Resultado do Teste
177.66.52.134	262488	4145	OPEN	2019-06-03T08:50:01Z	SMTP banner: confirmed
177.66.52.137	262488	4145	OPEN	2019-05-27T18:02:30Z	SMTP banner: confirmed
177.66.52.195	262488	4145	OPEN	2019-05-28T20:55:06Z	SMTP banner: confirmed
177.66.52.232	262488	4145	OPEN	2019-05-28T20:55:17Z	SMTP banner: confirmed
177.66.52.252	262488	4145	OPEN	2019-05-29T22:44:48Z	SMTP banner: confirmed
179.106.162.108	262488	4145	OPEN	2019-05-27T23:52:08Z	SMTP banner: confirmed
179.106.162.145	262488	4145	OPEN	2019-05-27T18:04:40Z	SMTP banner: confirmed
179.106.166.57	262488	4145	OPEN	2019-05-30T14:10:38Z	SMTP banner: confirmed
179.106.166.92	262488	4145	OPEN	2019-05-27T18:04:41Z	SMTP banner: confirmed

Cordialmente,

--

CERT.br/NIC.br

<cert@cert.br>

https://www.cert.br/