CONCEITOS E IMPLEMENTAÇÃO DE CGNAT

Marcelo Gondim
Fernando Frediani

O Brasil Peering Fórum

É um **NOG** (**Network Operators Group**) onde profissionais da área trabalham compartilhando conteúdo técnico com o objetivo de fazer uma Internet Brasileira melhor.

Possui uma Wiki aberta (**https://wiki.brasilpeeringforum.org/**) para facilitar o compartilhamento de conteúdo em forma de artigos e tutoriais e também uma Lista de Discussão (**https://listas.brasilpeeringforum.org/**).

Qualquer pessoa cadastrada pode contribuir com novos artigos ou tutoriais ou acrescentar informações complementares.

Acesso gratuito.

O Brasil Peering Fórum

Crie uma conta Entrar

Página principal | Discussão | Ler | Ver código-fonte | Ver histórico | Pesquisar em Wiki BPF

Página principal

Seja bem vindo à Wiki do BPF (Brasil Peering Forum).

O **Brasil Peering Forum** é um NOG (Network Operators Group) onde vários profissionais trabalham com o objetivo de fazer uma Internet Brasileira melhor. Engajados com a comunidade de operadores de Redes e Telecomunicações no Brasil, desempenham papéis instrutivos e participam nos principais eventos do setor, colaborando para o crescimento técnico e operacional dos ISP's e empresas da área de Internet. Contribuem de forma ativa com várias listas de discussão técnicas no Brasil e no mundo, e estão sempre abertos a um bom bate-papo sobre processos para elevar o nível dos ISP's nacionais.

A participação é aberta para a comunidade Internet e gratuita e acontece através dos grupos de trabalho e listas de discussão. Convidamos todos a se inscreverem e participar das discussões na **Lista Geral de Discusão BPF**. Ali são discutidos os assuntos de interesse geral, realizados anúncios para a comunidade, aviso de publicação de novos materiais, etc. O intuito principal da lista é promover a troca de informações, aprendizado e networking entre os participantes. Para se inscrever acesse a página sobre Participação (Listas de Discussão / Task-Forces).

Conheça os detalhes do trabalho desenvolvido pelo BPF nos links abaixo

Artigos em Destaque

Acesso rápido à artigos em destaque e de uso frequente.

- **Como Escrever na Wiki** - Passo a Passo de como criar um novo artigo e contribuir com a Wiki do BPF.
- **CDN Peering e PNI - Brasil** - Lista com as principais CDNs, instruções de como solicitar Servidores, sessões Bilaterias nos IXs e PNIs.

Últimos Artigos Publicados

Lista completa de todos os artigos e materiais publicados na área **Conteúdos**.

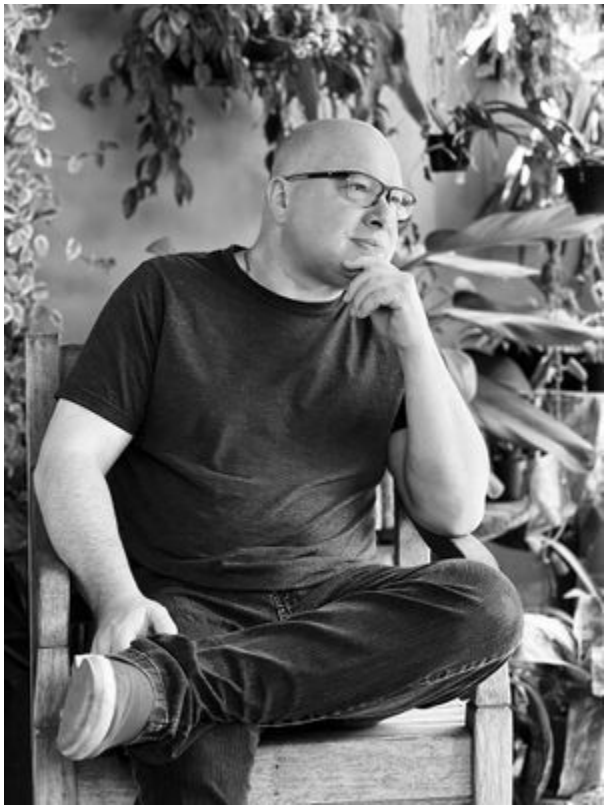
- **Assinatura MoU BPF** - Assinatura do Memorando de Entendimento entre os membros da Board e Comitê de Programa do BPF.
- **O Minimo que Voce precisa saber sobre IRR** - Artigo explicando o que é IRR, a importância do uso, principais bases e com um tutorial de como adicionar informações em uma base.
- **Informativo Infra 07** - 29/12/2019
- **Boas praticas para a implantação do OSPF em ambientes de ISP** - Artigo discorrendo sobre 12 boas práticas em situações envolvendo OSPF em ambientes ISP.
- **Introdução aos Conceitos de Programabilidade de Infraestruturas de Redes** - Artigo um tanto extenso e completo cobrindo os fundamentos de programabilidade de redes
- **Informativo Infra 08** - 26/01/2019
- **Informativo Infra 09** - 16/02/2020
- **Informativo Infra 10** - 09/03/2020
- **DNSSEC - Seguranca do DNS** - Artigo conceitual explicando o funcionando do DNSSEC baseado no documento DNSSEC: Securing DNS publicado pela ICANN.
- **UTRS - Registro e Configuração** - Artigo que explica o funcionamento do serviço UTRS do Team Cymru, passo a passo para solicitação e configurações exemplo.
- **Soluções para o Gerenciamento Efetivo do BGP em um Sistema Autônomo** - Artigo bastante completo dissertando sobre o gerenciamento e monitoramento do BGP em um Sistema Autônomo.

Página principal
Mudanças recentes
Página aleatória
Ajuda

Menu
Quem Somos
Participação
Conteúdos Úteis
Categorias
Documentos Públicos
Agenda

Ferramentas
Páginas afluentes
Mudanças relacionadas
Páginas especiais
Versão para impressão
Ligação permanente
Informações da página

Apresentadores - Marcelo Gondim



- Começou sua carreira como desenvolvedor de software em COBOL e Clipper entre 1992 e 1995. Em 1996 foi responsável por desenvolver um sistema concorrente com o RENPAC da Embratel para acesso ao SISCOMEX e implantou a Internet para fins comerciais na empresa DATABRAS.
- Trabalhou como consultor e instrutor de Linux na Conectiva S/A em 2000.
- Em 2003 se tornou consultor de diversos Provedores de Internet na Região dos Lagos - RJ e onde acabou se tornando CTO da Nettel Telecomunicações (AS53135) com 42.000 assinantes. Implantou IPv6 iniciando em 2013 e se tornou participante do MANRS com diversas contribuições com artigos e palestras.
- Atualmente é Especialista em Redes, cuida do SOC (Security Operations Center) da Brasil TecPar AS262907, onde desenvolve as boas práticas, tratamentos de incidentes relacionados ao ASN e desenvolve as estratégias de Mitigação DDoS da empresa. Também desenvolveu uma Rede de DNS Recursivo Anycast espalhada pelo RS e MT/MS, também certificada KINDNS.

Apresentadores - Fernando Frediani

Engenheiro de Computação graduado pela Pontifícia Universidade Católica de Campinas. Possui especialização pela Universidade de Cranfield no Reino Unido e MBA em Gestão Empresarial pela Fundação Getúlio Vargas no Brasil.

Na empresa Americanet/Ultrawave exerce o cargo de Gerente de Engenharia. Atuou como consultor de diversos Provedores de Serviços e Banda Larga com foco em Infraestrutura. Também exerceu a função de Gerente de Engenharia e Infraestrutura na empresa UPX Technologies, Systems Architect na NTT Europe e Lead Systems Engineer na

Qube Managed Services, ambas em Londres, Reino Unido.

Desenhou e implantou diversos projetos de Cloud e Infraestrutura como Serviço em países como Reino Unido, Estados Unidos, Espanha, França, Suíça e Alemanha.

Membro fundador e atualmente membro da Diretoria do Brasil Peering Fórum (https://wiki.brasilpeeringforum.org), participa também de diversos fóruns relacionados à Governança de Internet como Fórum de Políticas do LACNIC, ARIN e AfriNic. Foi aluno da Escola Brasileira de Governança da Internet (EGI.br)

Palestrante em eventos do setor de Internet no Brasil e no exterior.

B.P.F
Brasil Peering Fórum

Introdução

- Surgiu devido à escassez de IPv4 disponível para os Provedores de Acesso.
- Alocação de Endereços definida pela RFC6598.
  - Range 100.64.0.0/10.
  - Não é o mesmo que RFC1918.
- Uma maneira sustentável e organizada para continuar provendo acesso até a transição completa para IPv6.
- CGNAT é NAT.
- CGNAT “não é NAT”.

Aspectos Legais

- Importância do registro e guarda de logs para identificação do usuário.
    - Art. 10, Art. 13 e Art. 15 do Marco Civil.
- Somente o endereço IP de origem não é suficiente. É necessário haver o registro da porta de origem também.
    - Interpretações do Judiciário no sentido da obrigatoriedade da guarda também da porta de origem.
- Não se deve jamais registrar endereço de destino para este propósito (violação da privacidade).
- Provedores de conteúdo devem também guardar os registros de porta de origem, caso contrário a identificação não é possível.

Tipos de CGNAT - Determinístico

- Mais utilizado pelos provedores em geral pela facilidade de implementação.
- Define um range limitado de portas TCP e UDP por usuário para ser utilizado.
- Permite uma economia razoável de endereços IPv4 Públicos à depender do nível de compartilhamento realizado.
- Requer uma quantidade bem menor de log (apenas os de autenticação e atribuição do IP da range de CGNAT).

Tipos de CGNAT - Determinístico

- Exemplo 1 – Compartilhamento 1:32
  - 32 assinantes compartilham o mesmo IPv4 Público
  - 2016 portas de origem alocadas para cada IP Privado
- Exemplo 2 – Compartilhamento 1:16
  - 16 assinantes compartilham o mesmo IPv4 Público
  - 4032 portas de origem alocadas para cada IP Privado
- Exemplo 3 – Compartilhamento 1:8
  - 8 assinantes compartilharão o mesmo IPv4 Público
  - 8064 portas de origem alocadas para cada IP Privado

```
iptables -t nat -A CGNAT -s 100.64.18.10 -p tcp -j SNAT --to
192.0.0.1:3040-5055
iptables -t nat -A CGNAT -s 100.64.18.10 -p udp -j SNAT --to
192.0.0.1:3040-5055
```

Tipos de CGNAT – Bulk Port Allocation

- Realiza a atribuição de portas de origem para cada IP de CGNAT de maneira dinâmica e em blocos, conforme a necessidade de cada assinante.
- Define um range máximo de portas TCP e UDP inicial por usuário e blocos adicionais à serem alocados posteriormente conforme a necessidade de cada um.
- Permite uma economia maior de endereços IPv4 Públicos pois a maioria dos usuários não utilizam um alto número de portas e um mesmo IPv4 pode ser utilizado por uma quantidade maior de usuários.
- Requer registro de log devido às alocações serem realizadas de maneira dinâmica.

Tipos de CGNAT – Bulk Port Allocation

- Exemplo 1
  - Assinante recebe inicialmente uma alocação de 512 portas para uso.
  - Quando atingir uso do número de portas alocadas o sistema alocará blocos adicionais de 512 portas (não contíguas) para o mesmo IP Privado utilizado pelo assinante.
- Exemplo 2
  - Assinante recebe inicialmente uma alocação de 256 portas para uso.
  - Quando estiver perto de atingir uso do número de portas alocadas o sistema alocará blocos adicionais de 128 portas (não contíguas) para o mesmo IP Privado utilizado pelo assinante.
- Cada nova alocação gera uma entrada nos logs.

Processo de Identificação de Usuário

IP Destino: 200.160.2.3
Porta de Destino: 443
IP de Origem: 198.51.100.22
Porta de Origem: 48122

IP Destino: 200.160.2.3
Porta de Destino: 443
IP de Origem: 100.64.1.22
Porta de Origem: 21402

B . P . F
Brasil Peering Fórum

Processo de Identificação de Usuário

Log do Servidor

```
198.51.100.22 48122 - servidor-web [28/Apr/2023:10:22:44 -0300] "GET
/index.html HTTP/1.1" 201 1126 "-" ""Mozilla/5.0 (Windows NT 10.0; Win64;
x64; rv:59.0) Gecko/20100101 Firefox/59.0"
```

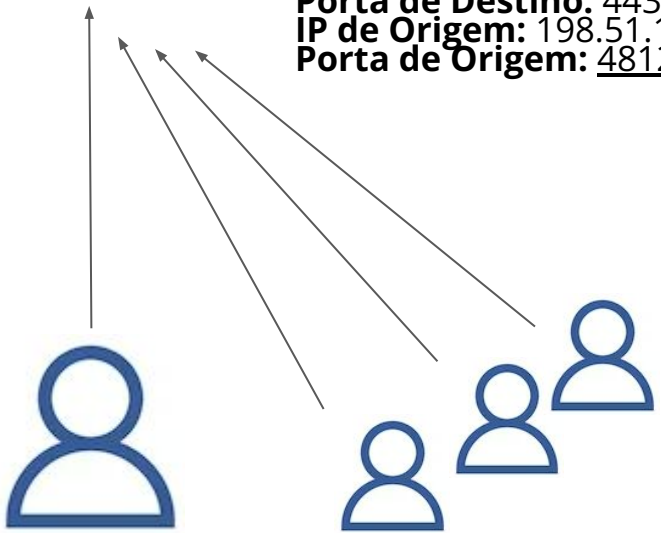
Processo de Identificação de Usuário

IP Destino: 200.160.2.3
Porta de Destino: 443
IP de Origem: 198.51.100.22
Porta de Origem: 23482

IP Destino: 200.160.2.3
Porta de Destino: 443
IP de Origem: 198.51.100.22
Porta de Origem: 18680

IP Destino: 200.160.2.3
Porta de Destino: 443
IP de Origem: 198.51.100.22
Porta de Origem: 48122

IP Destino: 200.160.2.3
Porta de Destino: 443
IP de Origem: 198.51.100.22
Porta de Origem: 14288



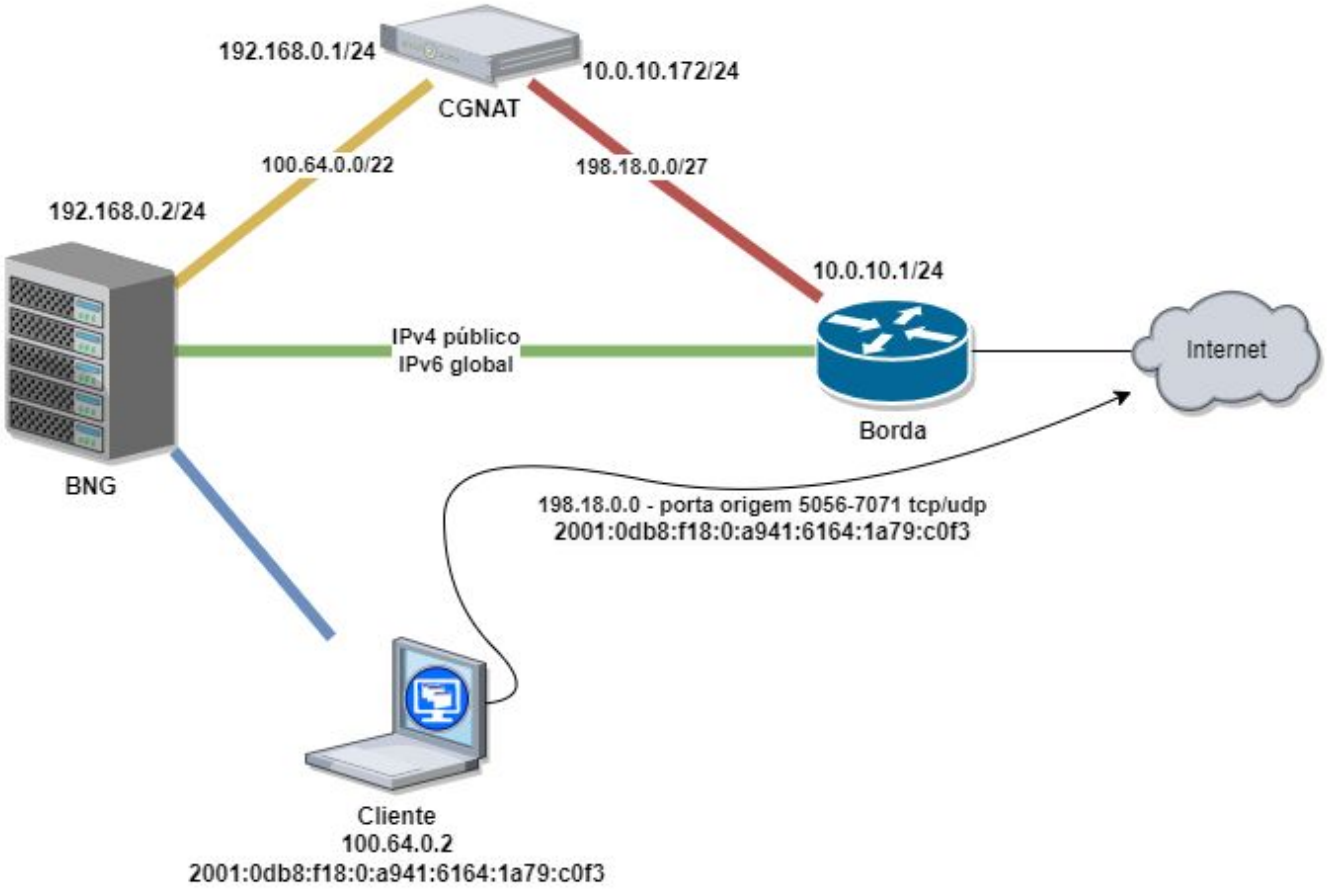
IP Destino: 200.160.2.3
Porta de Destino: 443
IP de Origem: 100.64.1.22
Porta de Origem: 21402

B . P . F
Brasil Peering Fórum

Processo de Identificação de Usuário

- **Informações necessárias serem fornecidas pelo Provedor de Conteúdo para identificação**
  - IP de Origem
  - Porta de Origem
  - Data e horário do acesso ao conteúdo (com fuso horário)
- Com essas informações é solicitado ao **Provedor de Acesso** que consulta seus registros.
  - No caso do **IP de Origem ser IP Público alocado para CGNAT** verifica seus registros/logs para identificar qual o IP Interno de CGNAT foi utilizado
    - Para isso ser possível é **necessária a informação da Porta de Origem**
  - Verifica-se então nos registros de autenticação qual usuário recebeu aquele IP naquela data e horário.

Diagrama Fluxo CGNAT Determinístico



B.P.F
Brasil Peering Fórum

CGNAT Determinístico 1/32 com GNU/Linux Debian 11 (Bullseye)

debian

B.P.F
Brasil Peering Fórum

Hardware e Sistema que utilizaremos neste tutorial:

- 2x Intel® Xeon® Silver 4215R Processor (3.20 GHz, 11M Cache, 8 núcleos/16 threads). Ambiente **NUMA** (**non-uniform memory access**).
- 32Gb de ram.
- 2x SSD 240 Gb **RAID1**.
- 2x Interfaces de rede **Intel XL710-QDA2** (**2 portas de 40 Gbps**).
- GNU/Linux Debian 11 (Bullseye).

- **LACP** - 2 portas de cada interface para redundância.

- Especificações PCIe de barramento por **versão** x **lane (x1)**:

  - PCIe 1.0/1.1 - 2.5 GT/s - (8b/10b encoding) - 2 Gbps.
  - PCIe 2.0/2.1 - 5.0 GT/s - (8b/10b encoding) - 4 Gbps.
  - PCIe 3.0/3.1 - 8.0 GT/s - (128b/130b encoding) - ~7,88 Gbps.
  - PCIe 4.0 - 16 GT/s - (128b/130b encoding) - ~15,76 Gbps.

Calculando a capacidade:

Se observarmos a **XL710-QDA2** é **PCIe 3.0 x8 (8 lanes)** ou seja o barramento irá suportar:

- 8.0 GT/s * (128b/130b encoding) * 8 lanes = **63,01 Gbps**

O objetivo do **LACP** nesse caso, não seria alcançar os **80 Gbps** de capacidade em cada interface, mesmo porque cada barramento das interfaces é limitado em **63,01 Gbps**, mas manteremos um backup dos **40 Gbps**.

- 2 interfaces: Tráfego **63,01 Gbps de entrada** e **63,01 Gbps de saída**.
- **CPU Affinity**.

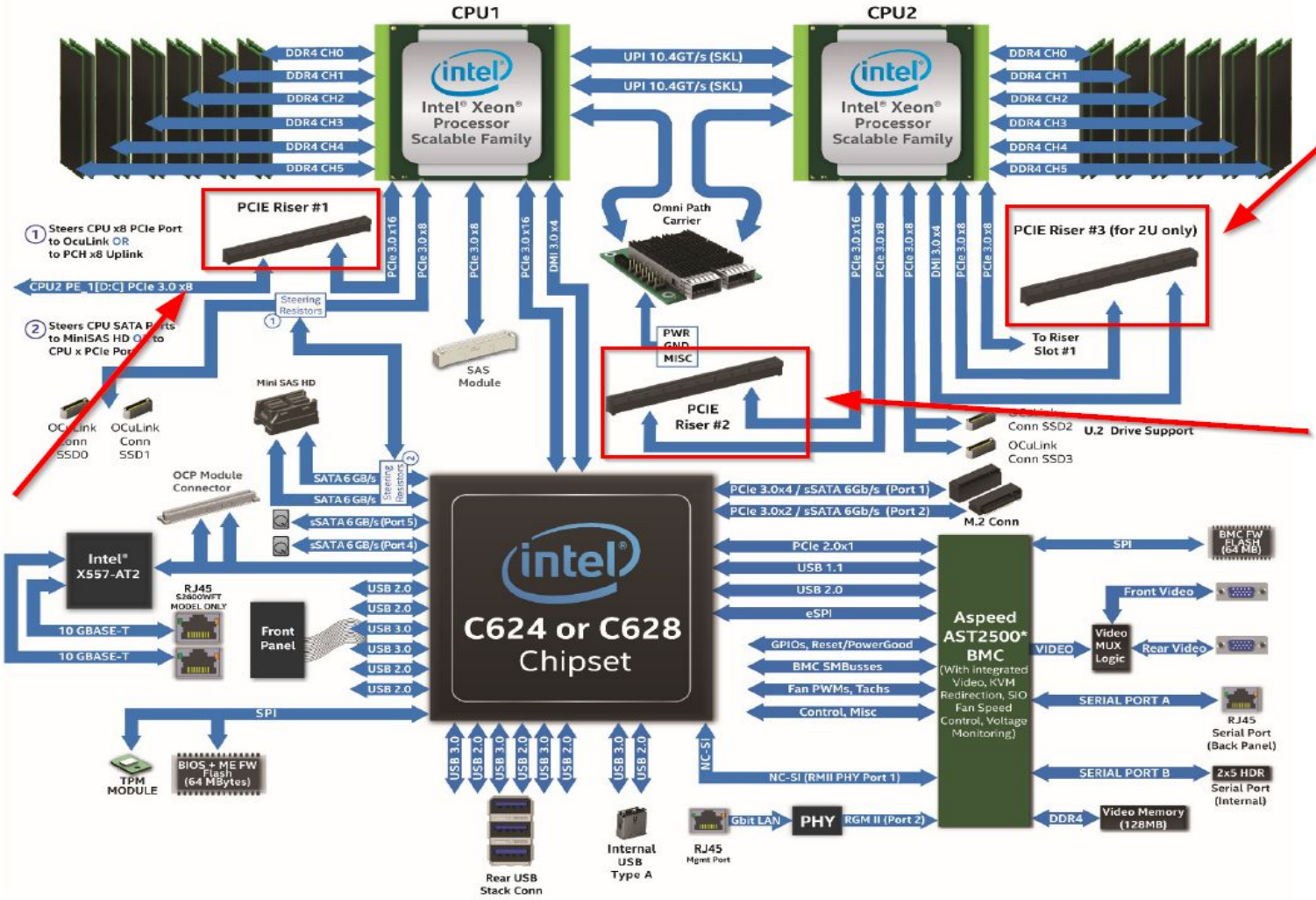
Teste para suporte a CPU Affinity:

Teste para suporte a CPU Affinity:

```
# cat /sys/class/net/<interface>/device/numa_node
```

- Resultado **-1**: sem suporte.
- Resultado **0**: controle CPU0.
- Resultado **1**: controle CPU1.

S2600WF Architecture Block Diagram



B.P.F
Brasil Peering Fórum

Cores por CPU:

```
# cat /sys/devices/system/node/node0/cpulist
0-7

# cat /sys/devices/system/node/node1/cpulist
8-15
```

No exemplo acima a **CPU0** tem os cores de **0** a **7** e a **CPU1**, os cores de **8** a **15**, ou seja, é um equipamento com **16 cores**.

Também é importante, para aumento de performance, que seja desabilitado na BIOS o **HT (Hyper Threading)**.

Ferramenta de tuning:

```
# apt install ethtool
```

Nosso exemplo temos **16 cores** sendo que **8 cores** por **CPU**.

```
# ethtool -L enp5s0f0 combined 8
# ethtool -L enp5s0f1 combined 8
# ethtool -L enp6s0f0 combined 8
# ethtool -L enp6s0f1 combined 8
```

CPU Affinity:

- Remova **irqbalance** e use ferramenta de **cpu affinity**:

Intel Network Adapter

```
# /root/scripts/set_irq_affinity 0-7 enp5s0f0
# /root/scripts/set_irq_affinity 0-7 enp5s0f1
# /root/scripts/set_irq_affinity 8-15 enp6s0f0
# /root/scripts/set_irq_affinity 8-15 enp6s0f1
```

- **Rings RX** e **TX**.

```
# ethtool -g enp5s0f0
Ring parameters for enp5s0f0:
Pre-set maximums:
RX:             4096
RX Mini:        n/a
RX Jumbo:       n/a
TX:             4096
Current hardware settings:
RX:             512
RX Mini:        n/a
RX Jumbo:       n/a
TX:             512
```

Acima vimos que o valor máximo é de **4096** tanto para **TX**, quanto para **RX** mas está configurado para **512** em **RX** e **TX**. Façamos então:

```
# ethtool -G enp5s0f0 rx 4096 tx 4096
# ethtool -G enp5s0f1 rx 4096 tx 4096
# ethtool -G enp6s0f0 rx 4096 tx 4096
# ethtool -G enp6s0f1 rx 4096 tx 4096
```

Vamos desabilitar as seguintes options das interfaces: **TSO**, **GRO** e **GSO**.

```
# ethtool -K enp5s0f0 tso off gro off gso off
# ethtool -K enp5s0f1 tso off gro off gso off
# ethtool -K enp6s0f0 tso off gro off gso off
# ethtool -K enp6s0f1 tso off gro off gso off
```

Aumentaremos o **txqueuelen** para **10000**:

```
# ip link set enp5s0f0 txqueuelen 10000
# ip link set enp5s0f1 txqueuelen 10000
# ip link set enp6s0f0 txqueuelen 10000
# ip link set enp6s0f1 txqueuelen 10000
```

Habilitando bonding:

```
# apt install ifenslave
# modprobe bonding
# echo "bonding" >> /etc/modules
```

Salvaremos tudo o que fizemos em **/etc/network/interfaces**, configurado conforme nosso diagrama, usando **LACP** e executando nossos comandos anteriores.

```
auto bond0
iface bond0 inet static
        bond-slaves enp5s0f0 enp5s0f1
        bond_mode 802.3ad
        bond-ad_select bandwidth
        bond_miimon 100
        bond_downdelay 200
        bond_updelay 200
        bond-lacp-rate 1
        bond-xmit-hash-policy layer2+3
        address 10.0.10.172/24
        gateway 10.0.10.1
        pre-up /usr/sbin/ethtool -L enp5s0f0 combined 8
        pre-up /usr/sbin/ethtool -L enp5s0f1 combined 8
        pre-up /root/scripts/set_irq_affinity 0-7 enp5s0f0
        pre-up /root/scripts/set_irq_affinity 0-7 enp5s0f1
        pre-up /usr/sbin/ethtool -G enp5s0f0 rx 4096 tx 4096
        pre-up /usr/sbin/ethtool -G enp5s0f1 rx 4096 tx 4096
        pre-up /usr/sbin/ethtool -K enp5s0f0 tso off gro off gso off
        pre-up /usr/sbin/ethtool -K enp5s0f1 tso off gro off gso off
        pre-up /usr/sbin/ip link set enp5s0f0 txqueuelen 10000
        pre-up /usr/sbin/ip link set enp5s0f1 txqueuelen 10000
```

B.P.F

Brasil Peering Fórum

```
auto bond1
iface bond1 inet static
        bond-slaves enp6s0f0 enp6s0f1
        bond_mode 802.3ad
        bond-ad_select bandwidth
        bond_miimon 100
        bond_downdelay 200
        bond_updelay 200
        bond-lacp-rate 1
        bond-xmit-hash-policy layer2+3
        address 192.168.0.1/24
        pre-up /usr/sbin/ethtool -L enp6s0f0 combined 8
        pre-up /usr/sbin/ethtool -L enp6s0f1 combined 8
        pre-up /root/scripts/set_irq_affinity 8-15 enp6s0f0
        pre-up /root/scripts/set_irq_affinity 8-15 enp6s0f1
        pre-up /usr/sbin/ethtool -G enp6s0f0 rx 4096 tx 4096
        pre-up /usr/sbin/ethtool -G enp6s0f1 rx 4096 tx 4096
        pre-up /usr/sbin/ethtool -K enp6s0f0 tso off gro off gso off
        pre-up /usr/sbin/ethtool -K enp6s0f1 tso off gro off gso off
        pre-up /usr/sbin/ip link set enp6s0f0 txqueuelen 10000
        pre-up /usr/sbin/ip link set enp6s0f1 txqueuelen 10000
```

Colocaremos o **kernel do backports**. Para isso deixe o seu **/etc/apt/sources** conforme abaixo e rode os comandos na sequência:

```
deb http://security.debian.org/debian-security bullseye-security main contrib non-free
deb http://deb.debian.org/debian bullseye main non-free contrib
deb http://deb.debian.org/debian bullseye-updates main contrib non-free
deb http://deb.debian.org/debian bullseye-backports main contrib non-free
```

```
# apt update
# apt install -t bullseye-backports linux-image-amd64
# reboot
```

static loop e rc.local:

Recomendações sobre Mitigação DDoS

```
# > /etc/rc.local
# chmod +x /etc/rc.local
```

Dentro teremos:

```
#!/bin/sh -e
/usr/sbin/ip route add blackhole 198.18.0.0/27 metric 254
/usr/sbin/route add -net 100.64.0.0/22 gw 192.168.0.2
```

No exemplo acima estamos colocando em **blackhole** o nosso prefixo IPv4 público deste tutorial que é o **198.18.0.0/27** e adicionando uma rota de retorno do prefixo **100.64.0** usado no nosso BNG para o **next-hop 192.168.0.2**.

B . P . F
Brasil Peering Fórum

Redução dos tempos de timeouts tcp e udp:

- No **/etc/rc.local**:

```
echo 5 > /proc/sys/net/netfilter/nf_conntrack_tcp_timeout_syn_sent
echo 5 > /proc/sys/net/netfilter/nf_conntrack_tcp_timeout_syn_recv
echo 86400 > /proc/sys/net/netfilter/nf_conntrack_tcp_timeout_established
echo 10 > /proc/sys/net/netfilter/nf_conntrack_tcp_timeout_fin_wait
echo 10 > /proc/sys/net/netfilter/nf_conntrack_tcp_timeout_close_wait
echo 10 > /proc/sys/net/netfilter/nf_conntrack_tcp_timeout_last_ack
echo 10 > /proc/sys/net/netfilter/nf_conntrack_tcp_timeout_time_wait
echo 10 > /proc/sys/net/netfilter/nf_conntrack_tcp_timeout_close
echo 300 > /proc/sys/net/netfilter/nf_conntrack_tcp_timeout_max_retrans
echo 300 > /proc/sys/net/netfilter/nf_conntrack_tcp_timeout_unacknowledged
echo 10 > /proc/sys/net/netfilter/nf_conntrack_udp_timeout
echo 180 > /proc/sys/net/netfilter/nf_conntrack_udp_timeout_stream
echo 10 > /proc/sys/net/netfilter/nf_conntrack_icmp_timeout
echo 600 > /proc/sys/net/netfilter/nf_conntrack_generic_timeout
```

Tuning /etc/sysctl.conf:

```
net.core.default_qdisc=fq
net.ipv4.tcp_congestion_control=bbr
net.core.rmem_max = 2147483647
net.core.wmem_max = 2147483647
net.ipv4.tcp_rmem = 4096 87380 2147483647
net.ipv4.tcp_wmem = 4096 65536 2147483647
net.ipv4.conf.all.forwarding=1
net.netfilter.nf_conntrack_helper=1
net.netfilter.nf_conntrack_buckets = 512000
net.netfilter.nf_conntrack_max = 4096000
vm.swappiness=10
```

- Contador conntracks em uso:

```
# cat /proc/sys/net/netfilter/nf_conntrack_count
```

- Listar as conntracks:

```
# cat /proc/net/nf_conntrack
```

Ajustando a data e horário do sistema:

```
# apt install chrony
```

```
# cat << EOF > /etc/chrony/chrony.conf
confdir /etc/chrony/conf.d
sourcedir /run/chrony-dhcp
sourcedir /etc/chrony/sources.d
keyfile /etc/chrony/chrony.keys
driftfile /var/lib/chrony/chrony.drift
ntsdumpdir /var/lib/chrony
logdir /var/log/chrony
maxupdateskew 100.0
rtcsync
makestep 1 3
leapsectz right/UTC
EOF
```

Ajustando a data e horário do sistema:

```
# cat << EOF > /etc/chrony/sources.d/nic.sources
server a.st1.ntp.br iburst nts
server b.st1.ntp.br iburst nts
server c.st1.ntp.br iburst nts
server d.st1.ntp.br iburst nts
EOF
```

```
# systemctl restart chronyd.service
# timedatectl set-timezone "UTC"
```

ALGs (Application Layer Gateway):

Em **/etc/modules** adicionaremos mais os módulos abaixo:

```
nf_conntrack
nf_nat_pptp
nf_nat_h323
nf_nat_sip
nf_nat_irc
nf_nat_ftp
nf_nat_tftp
```

Iniciando regras CGNAT:

Antes de começarmos nossas regras de CGNAT precisaremos de alguns pacotes:

```
# apt install python3-pip nftables
# pip install ipaddress
```

Gerador de regras de CGNAT para **nftables**: **GRCN (José Beiriz).**

Nosso script será dividido em 2 partes:

- O script base **/root/scripts/frw-nft.sh**.
- Parte composta pelos arquivos de regras de CGNAT, onde são feitas as traduções de IPs privados **100.6** **(Shared Address Space - RFC6598)**, para os IPs públicos.

B . P . F

Brasil Peering Fórum

Nosso script de CGNAT base **/root/scripts/frw-nft.sh**:

```
#!/usr/sbin/nft -f
# limpa todas as regras da memoria
flush ruleset

# regras base para o CGNAT
add table ip nat
add chain ip nat POSTROUTING { type nat hook postrouting priority 100; policy accept; }
add chain ip nat CGNATOUT

# libera o proprio CGNAT para acessar a Internet - para atualizacoes por exemplo
add rule ip nat POSTROUTING oifname "bond0" ip saddr 10.0.10.172 counter snat to
198.18.0.0

# faz o jump para as regras de CGNAT
add rule ip nat POSTROUTING oifname "bond0" counter jump CGNATOUT

# carrega os arquivos de regras de CGNAT
include "/root/scripts/cgnat-0-31.conf"
```

Após a criação do script, alteramos a permissão dele para ficar como executável e adicionamos ele em nosso **/etc/rc.local**:

```
# chmod 700 /root/scripts/frw-nft.sh
# echo "/root/scripts/frw-nft.sh" >> /etc/rc.local
```

- Gerando nossas regras de CGNAT:

Como estamos trabalhando no modelo **determinístico de 1/32**, basta pegarmos nosso **bloco privado 100.64.0.0/22 (1024 IPs)** e nosso **bloco público 198.18.0.0/27 (32 IPs)** e executarmos em linha de comando:

```
# cd /root/scripts
# ./cgnat-nft.py 0 198.18.0.0/27 100.64.0.0/22 1/32
```

- No comando anterior temos o número **0** como índice inicial e sendo 1/32, o programa calcula automaticamente o índice final. Será gerado o arquivo **cgnat-0-31.conf**.

- Se for gerar novas regras, o próximo índice a ser usado seria o 32. Por exemplo:

```
# ./cgnat-nft.py 32 198.18.0.32/27 100.64.4.0/22 1/32
```

- O comando acima irá gerar novas regras no arquivo **cgnat-32-63.conf**.

Executando o gerador de regras:

```
./cgnat-nft.py 0 198.18.0.0/27 100.64.0.0/22 1/32
```

```
################################################################################################
    GRCN - Gerador de Regras CGNAT em nftables - Beiriz - v4.0 - 27/07/2020 (25/03/2023)
################################################################################################

        [ Índice inicial: 0 | público: 198.18.0.0/27 | privado: 100.64.0.0/22 | 2016 portas/IP (1/32)]

 - Índice das regras: 0;
 - Rede pública: 198.18.0.0/27 (32 IPs);
 - Rede privada: 100.64.0.0/22 (1024 IPs);
 - Quantidade de IPs privados por IP público: 32 (32 sub-redes /27);
 - Total de portas públicas: 64512;
 - Portas por IP privado: 2016;
 - Arquivo de destino (conf): 'cgnat-0-31.conf';


Tecle [ENTER]...
```

B.P.F
Brasil Peering Fórum

```
# GRCN - Gerador de Regras CGNAT em nftables - Beiriz - v4.0 - 27/07/2020 (25/03/2023)
# - blocos 100.64.0.0/22 -> 198.18.0.0/27;
# - /0 de IPs privados / IP público;
# - 2016 portas / IP privado;
# ------------------------------------ #INDICE 0 / IP PUBLICO 198.18.0.0
add chain ip nat CGNATOUT_0
flush chain ip nat CGNATOUT_0
add rule ip nat CGNATOUT_0 ip protocol tcp ip saddr 100.64.0.0 counter snat to 198.18.0.0:1024-3039
add rule ip nat CGNATOUT_0 ip protocol udp ip saddr 100.64.0.0 counter snat to 198.18.0.0:1024-3039
add rule ip nat CGNATOUT_0 ip protocol tcp ip saddr 100.64.0.1 counter snat to 198.18.0.0:3040-5055
add rule ip nat CGNATOUT_0 ip protocol udp ip saddr 100.64.0.1 counter snat to 198.18.0.0:3040-5055
add rule ip nat CGNATOUT_0 ip protocol tcp ip saddr 100.64.0.2 counter snat to 198.18.0.0:5056-7071
add rule ip nat CGNATOUT_0 ip protocol udp ip saddr 100.64.0.2 counter snat to 198.18.0.0:5056-7071
add rule ip nat CGNATOUT_0 ip protocol tcp ip saddr 100.64.0.3 counter snat to 198.18.0.0:7072-9087
add rule ip nat CGNATOUT_0 ip protocol udp ip saddr 100.64.0.3 counter snat to 198.18.0.0:7072-9087
add rule ip nat CGNATOUT_0 ip protocol tcp ip saddr 100.64.0.4 counter snat to 198.18.0.0:9088-11103
add rule ip nat CGNATOUT_0 ip protocol udp ip saddr 100.64.0.4 counter snat to 198.18.0.0:9088-11103
add rule ip nat CGNATOUT_0 ip protocol tcp ip saddr 100.64.0.5 counter snat to 198.18.0.0:11104-13119
add rule ip nat CGNATOUT_0 ip protocol udp ip saddr 100.64.0.5 counter snat to 198.18.0.0:11104-13119
add rule ip nat CGNATOUT_0 ip protocol tcp ip saddr 100.64.0.6 counter snat to 198.18.0.0:13120-15135
add rule ip nat CGNATOUT_0 ip protocol udp ip saddr 100.64.0.6 counter snat to 198.18.0.0:13120-15135
add rule ip nat CGNATOUT_0 ip protocol tcp ip saddr 100.64.0.7 counter snat to 198.18.0.0:15136-17151
add rule ip nat CGNATOUT_0 ip protocol udp ip saddr 100.64.0.7 counter snat to 198.18.0.0:15136-17151
add rule ip nat CGNATOUT_0 ip protocol tcp ip saddr 100.64.0.8 counter snat to 198.18.0.0:17152-19167
add rule ip nat CGNATOUT_0 ip protocol udp ip saddr 100.64.0.8 counter snat to 198.18.0.0:17152-19167
add rule ip nat CGNATOUT_0 ip protocol tcp ip saddr 100.64.0.9 counter snat to 198.18.0.0:19168-21183
add rule ip nat CGNATOUT_0 ip protocol udp ip saddr 100.64.0.9 counter snat to 198.18.0.0:19168-21183
add rule ip nat CGNATOUT_0 ip protocol tcp ip saddr 100.64.0.10 counter snat to 198.18.0.0:21184-23199
add rule ip nat CGNATOUT_0 ip protocol udp ip saddr 100.64.0.10 counter snat to 198.18.0.0:21184-23199
add rule ip nat CGNATOUT_0 ip protocol tcp ip saddr 100.64.0.11 counter snat to 198.18.0.0:23200-25215
add rule ip nat CGNATOUT_0 ip protocol udp ip saddr 100.64.0.11 counter snat to 198.18.0.0:23200-25215
add rule ip nat CGNATOUT_0 ip protocol tcp ip saddr 100.64.0.12 counter snat to 198.18.0.0:25216-27231
add rule ip nat CGNATOUT_0 ip protocol udp ip saddr 100.64.0.12 counter snat to 198.18.0.0:25216-27231
add rule ip nat CGNATOUT_0 ip protocol tcp ip saddr 100.64.0.13 counter snat to 198.18.0.0:27232-29247
add rule ip nat CGNATOUT_0 ip protocol udp ip saddr 100.64.0.13 counter snat to 198.18.0.0:27232-29247
add rule ip nat CGNATOUT_0 ip protocol tcp ip saddr 100.64.0.14 counter snat to 198.18.0.0:29248-31263
add rule ip nat CGNATOUT_0 ip protocol udp ip saddr 100.64.0.14 counter snat to 198.18.0.0:29248-31263
add rule ip nat CGNATOUT_0 ip protocol tcp ip saddr 100.64.0.15 counter snat to 198.18.0.0:31264-33279
add rule ip nat CGNATOUT_0 ip protocol udp ip saddr 100.64.0.15 counter snat to 198.18.0.0:31264-33279
add rule ip nat CGNATOUT_0 ip protocol tcp ip saddr 100.64.0.16 counter snat to 198.18.0.0:33280-35295
"cgnat-0-31.conf" 2212L, 223100B
```

```
add rule ip nat CGNATOUT_0 ip protocol udp ip saddr 100.64.0.18 counter snat to 198.18.0.0:37312-39327
add rule ip nat CGNATOUT_0 ip protocol tcp ip saddr 100.64.0.19 counter snat to 198.18.0.0:39328-41343
add rule ip nat CGNATOUT_0 ip protocol udp ip saddr 100.64.0.19 counter snat to 198.18.0.0:39328-41343
add rule ip nat CGNATOUT_0 ip protocol tcp ip saddr 100.64.0.20 counter snat to 198.18.0.0:41344-43359
add rule ip nat CGNATOUT_0 ip protocol udp ip saddr 100.64.0.20 counter snat to 198.18.0.0:41344-43359
add rule ip nat CGNATOUT_0 ip protocol tcp ip saddr 100.64.0.21 counter snat to 198.18.0.0:43360-45375
add rule ip nat CGNATOUT_0 ip protocol udp ip saddr 100.64.0.21 counter snat to 198.18.0.0:43360-45375
add rule ip nat CGNATOUT_0 ip protocol tcp ip saddr 100.64.0.22 counter snat to 198.18.0.0:45376-47391
add rule ip nat CGNATOUT_0 ip protocol udp ip saddr 100.64.0.22 counter snat to 198.18.0.0:45376-47391
add rule ip nat CGNATOUT_0 ip protocol tcp ip saddr 100.64.0.23 counter snat to 198.18.0.0:47392-49407
add rule ip nat CGNATOUT_0 ip protocol udp ip saddr 100.64.0.23 counter snat to 198.18.0.0:47392-49407
add rule ip nat CGNATOUT_0 ip protocol tcp ip saddr 100.64.0.24 counter snat to 198.18.0.0:49408-51423
add rule ip nat CGNATOUT_0 ip protocol udp ip saddr 100.64.0.24 counter snat to 198.18.0.0:49408-51423
add rule ip nat CGNATOUT_0 ip protocol tcp ip saddr 100.64.0.25 counter snat to 198.18.0.0:51424-53439
add rule ip nat CGNATOUT_0 ip protocol udp ip saddr 100.64.0.25 counter snat to 198.18.0.0:51424-53439
add rule ip nat CGNATOUT_0 ip protocol tcp ip saddr 100.64.0.26 counter snat to 198.18.0.0:53440-55455
add rule ip nat CGNATOUT_0 ip protocol udp ip saddr 100.64.0.26 counter snat to 198.18.0.0:53440-55455
add rule ip nat CGNATOUT_0 ip protocol tcp ip saddr 100.64.0.27 counter snat to 198.18.0.0:55456-57471
add rule ip nat CGNATOUT_0 ip protocol udp ip saddr 100.64.0.27 counter snat to 198.18.0.0:55456-57471
add rule ip nat CGNATOUT_0 ip protocol tcp ip saddr 100.64.0.28 counter snat to 198.18.0.0:57472-59487
add rule ip nat CGNATOUT_0 ip protocol udp ip saddr 100.64.0.28 counter snat to 198.18.0.0:57472-59487
add rule ip nat CGNATOUT_0 ip protocol tcp ip saddr 100.64.0.29 counter snat to 198.18.0.0:59488-61503
add rule ip nat CGNATOUT_0 ip protocol udp ip saddr 100.64.0.29 counter snat to 198.18.0.0:59488-61503
add rule ip nat CGNATOUT_0 ip protocol tcp ip saddr 100.64.0.30 counter snat to 198.18.0.0:61504-63519
add rule ip nat CGNATOUT_0 ip protocol udp ip saddr 100.64.0.30 counter snat to 198.18.0.0:61504-63519
add rule ip nat CGNATOUT_0 ip protocol tcp ip saddr 100.64.0.31 counter snat to 198.18.0.0:63520-65535
add rule ip nat CGNATOUT_0 ip protocol udp ip saddr 100.64.0.31 counter snat to 198.18.0.0:63520-65535
add rule ip nat CGNATOUT_0 counter snat to 198.18.0.0
add rule ip nat CGNATOUT ip saddr 100.64.0.0/27 counter jump CGNATOUT_0
# ---------------------------------------- #INDICE 1 / IP PUBLICO 198.18.0.1
add chain ip nat CGNATOUT_1
flush chain ip nat CGNATOUT_1
add rule ip nat CGNATOUT_1 ip protocol tcp ip saddr 100.64.0.32 counter snat to 198.18.0.1:1024-3039
add rule ip nat CGNATOUT_1 ip protocol udp ip saddr 100.64.0.32 counter snat to 198.18.0.1:1024-3039
add rule ip nat CGNATOUT_1 ip protocol tcp ip saddr 100.64.0.33 counter snat to 198.18.0.1:3040-5055
add rule ip nat CGNATOUT_1 ip protocol udp ip saddr 100.64.0.33 counter snat to 198.18.0.1:3040-5055
add rule ip nat CGNATOUT_1 ip protocol tcp ip saddr 100.64.0.34 counter snat to 198.18.0.1:5056-7071
add rule ip nat CGNATOUT_1 ip protocol udp ip saddr 100.64.0.34 counter snat to 198.18.0.1:5056-7071
add rule ip nat CGNATOUT_1 ip protocol tcp ip saddr 100.64.0.35 counter snat to 198.18.0.1:7072-9087
add rule ip nat CGNATOUT_1 ip protocol udp ip saddr 100.64.0.35 counter snat to 198.18.0.1:7072-9087
```

```
table ip nat {
        chain POSTROUTING {
                type nat hook postrouting priority srcnat; policy accept;
                oifname "ens18" ip saddr 10.0.10.172 counter packets 0 bytes 0 snat to 198.18.0.0
                oifname "ens18" counter packets 0 bytes 0 jump CGNATOUT
        }

        chain CGNATOUT {
                ip saddr 100.64.0.0/27 counter packets 0 bytes 0 jump CGNATOUT_0
                ip saddr 100.64.0.32/27 counter packets 0 bytes 0 jump CGNATOUT_1
                ip saddr 100.64.0.64/27 counter packets 0 bytes 0 jump CGNATOUT_2
                ip saddr 100.64.0.96/27 counter packets 0 bytes 0 jump CGNATOUT_3
                ip saddr 100.64.0.128/27 counter packets 0 bytes 0 jump CGNATOUT_4
                ip saddr 100.64.0.160/27 counter packets 0 bytes 0 jump CGNATOUT_5
                ip saddr 100.64.0.192/27 counter packets 0 bytes 0 jump CGNATOUT_6
                ip saddr 100.64.0.224/27 counter packets 0 bytes 0 jump CGNATOUT_7
                ip saddr 100.64.1.0/27 counter packets 0 bytes 0 jump CGNATOUT_8
                ip saddr 100.64.1.32/27 counter packets 0 bytes 0 jump CGNATOUT_9
                ip saddr 100.64.1.64/27 counter packets 0 bytes 0 jump CGNATOUT_10
                ip saddr 100.64.1.96/27 counter packets 0 bytes 0 jump CGNATOUT_11
                ip saddr 100.64.1.128/27 counter packets 0 bytes 0 jump CGNATOUT_12
                ip saddr 100.64.1.160/27 counter packets 0 bytes 0 jump CGNATOUT_13
                ip saddr 100.64.1.192/27 counter packets 0 bytes 0 jump CGNATOUT_14
                ip saddr 100.64.1.224/27 counter packets 0 bytes 0 jump CGNATOUT_15
                ip saddr 100.64.2.0/27 counter packets 0 bytes 0 jump CGNATOUT_16
                ip saddr 100.64.2.32/27 counter packets 0 bytes 0 jump CGNATOUT_17
                ip saddr 100.64.2.64/27 counter packets 0 bytes 0 jump CGNATOUT_18
                ip saddr 100.64.2.96/27 counter packets 0 bytes 0 jump CGNATOUT_19
                ip saddr 100.64.2.128/27 counter packets 0 bytes 0 jump CGNATOUT_20
                ip saddr 100.64.2.160/27 counter packets 0 bytes 0 jump CGNATOUT_21
                ip saddr 100.64.2.192/27 counter packets 0 bytes 0 jump CGNATOUT_22
                ip saddr 100.64.2.224/27 counter packets 0 bytes 0 jump CGNATOUT_23
                ip saddr 100.64.3.0/27 counter packets 0 bytes 0 jump CGNATOUT_24
                ip saddr 100.64.3.32/27 counter packets 0 bytes 0 jump CGNATOUT_25
                ip saddr 100.64.3.64/27 counter packets 0 bytes 0 jump CGNATOUT_26
                ip saddr 100.64.3.96/27 counter packets 0 bytes 0 jump CGNATOUT_27
                ip saddr 100.64.3.128/27 counter packets 0 bytes 0 jump CGNATOUT_28
                ip saddr 100.64.3.160/27 counter packets 0 bytes 0 jump CGNATOUT_29
                ip saddr 100.64.3.192/27 counter packets 0 bytes 0 jump CGNATOUT_30
                ip saddr 100.64.3.224/27 counter packets 0 bytes 0 jump CGNATOUT_31
                ip saddr 100.64.4.0/27 counter packets 0 bytes 0 jump CGNATOUT_32
--Mais--
```

POC (Proof of Concept):

- **Proxmox** 3 VMs: **CGNAT**, **BNG** e **CLIENTE**.

- Do router de testes capturei os pacotes para demonstrar como funciona o **CGNAT** e a identificação do cliente em caso de recebimento de Ofício com os logs para busca.

No próximo slide teremos o acesso por parte do cliente e a captura dos pacotes somente para uma POC (Proof of Concept), para demonstrarmos que o CGNAT está funcionando e alocando a porta, dentro do range de portas, corretamente para um determinado cliente.

IP: 198.18.0.0
Porta origem: 6767

Packets Captured

```
08:27:17.529751 IP 198.18.0.0.6767 > 200.147.41.220.443: tcp 0
08:27:17.530652 IP 198.18.0.0.6767 > 200.147.41.220.443: tcp 85
08:27:17.540885 IP 200.147.41.220.443 > 198.18.0.0.6767: tcp 43
08:27:17.542010 IP 198.18.0.0.6767 > 200.147.41.220.443: tcp 788
08:27:17.558616 IP 200.147.41.220.443 > 198.18.0.0.6767: tcp 1408
08:27:17.558636 IP 200.147.41.220.443 > 198.18.0.0.6767: tcp 13
08:27:17.558747 IP 200.147.41.220.443 > 198.18.0.0.6767: tcp 1408
08:27:17.558768 IP 200.147.41.220.443 > 198.18.0.0.6767: tcp 13
08:27:17.558876 IP 200.147.41.220.443 > 198.18.0.0.6767: tcp 1408
08:27:17.558897 IP 200.147.41.220.443 > 198.18.0.0.6767: tcp 1408
08:27:17.558917 IP 200.147.41.220.443 > 198.18.0.0.6767: tcp 1408
08:27:17.558937 IP 200.147.41.220.443 > 198.18.0.0.6767: tcp 39
08:27:17.558998 IP 200.147.41.220.443 > 198.18.0.0.6767: tcp 1196
08:27:17.559604 IP 198.18.0.0.6767 > 200.147.41.220.443: tcp 0
08:27:17.559738 IP 198.18.0.0.6767 > 200.147.41.220.443: tcp 0
08:27:17.559756 IP 198.18.0.0.6767 > 200.147.41.220.443: tcp 0
08:27:20.635644 IP 198.18.0.0.6767 > 200.147.41.220.443: tcp 0
08:27:20.645147 IP 200.147.41.220.443 > 198.18.0.0.6767: tcp 0
08:27:20.645922 IP 198.18.0.0.6767 > 200.147.41.220.443: tcp 0
```

Exemplo Ofício: **198.18.0.0, porta 6767 em 20/03/2023 às 08:27:17 UTC-3**.
Busca no Radius: **100.64.0.2** em **20/03/2023 às 08:27:17 UTC-3**.

```
# GRCN - Gerador de Regras CGNAT em nftables - Beiriz - v4.001 - 27/07/2020 (31/03/2023)
# - blocos 100.64.0.0/22 -> 198.18.0.0/27;
# - /0 de IPs privados / IP público;
# - 2016 portas / IP privado;
# ------------------------------------ #INDICE 0 / IP PUBLICO 198.18.0.0
add chain ip nat CGNATOUT_0
flush chain ip nat CGNATOUT_0
add rule ip nat CGNATOUT_0 ip protocol tcp ip saddr 100.64.0.0 counter snat to 198.18.0.0:1024-3039
add rule ip nat CGNATOUT_0 ip protocol udp ip saddr 100.64.0.0 counter snat to 198.18.0.0:1024-3039
add rule ip nat CGNATOUT_0 ip protocol tcp ip saddr 100.64.0.1 counter snat to 198.18.0.0:3040-5055
add rule ip nat CGNATOUT_0 ip protocol udp ip saddr 100.64.0.1 counter snat to 198.18.0.0:3040-5055
add rule ip nat CGNATOUT_0 ip protocol tcp ip saddr 100.64.0.2 counter snat to 198.18.0.0:5056-7071
add rule ip nat CGNATOUT_0 ip protocol udp ip saddr 100.64.0.2 counter snat to 198.18.0.0:5056-7071
add rule ip nat CGNATOUT_0 ip protocol tcp ip saddr 100.64.0.3 counter snat to 198.18.0.0:7072-9087
add rule ip nat CGNATOUT_0 ip protocol udp ip saddr 100.64.0.3 counter snat to 198.18.0.0:7072-9087
add rule ip nat CGNATOUT_0 ip protocol tcp ip saddr 100.64.0.4 counter snat to 198.18.0.0:9088-11103
add rule ip nat CGNATOUT_0 ip protocol udp ip saddr 100.64.0.4 counter snat to 198.18.0.0:9088-11103
add rule ip nat CGNATOUT_0 ip protocol tcp ip saddr 100.64.0.5 counter snat to 198.18.0.0:11104-13119
add rule ip nat CGNATOUT_0 ip protocol udp ip saddr 100.64.0.5 counter snat to 198.18.0.0:11104-13119
add rule ip nat CGNATOUT_0 ip protocol tcp ip saddr 100.64.0.6 counter snat to 198.18.0.0:13120-15135
add rule ip nat CGNATOUT_0 ip protocol udp ip saddr 100.64.0.6 counter snat to 198.18.0.0:13120-15135
add rule ip nat CGNATOUT_0 ip protocol tcp ip saddr 100.64.0.7 counter snat to 198.18.0.0:15136-17151
add rule ip nat CGNATOUT_0 ip protocol udp ip saddr 100.64.0.7 counter snat to 198.18.0.0:15136-17151
add rule ip nat CGNATOUT_0 ip protocol tcp ip saddr 100.64.0.8 counter snat to 198.18.0.0:17152-19167
add rule ip nat CGNATOUT_0 ip protocol udp ip saddr 100.64.0.8 counter snat to 198.18.0.0:17152-19167
add rule ip nat CGNATOUT_0 ip protocol tcp ip saddr 100.64.0.9 counter snat to 198.18.0.0:19168-21183
add rule ip nat CGNATOUT_0 ip protocol udp ip saddr 100.64.0.9 counter snat to 198.18.0.0:19168-21183
add rule ip nat CGNATOUT_0 ip protocol tcp ip saddr 100.64.0.10 counter snat to 198.18.0.0:21184-23199
add rule ip nat CGNATOUT_0 ip protocol udp ip saddr 100.64.0.10 counter snat to 198.18.0.0:21184-23199
add rule ip nat CGNATOUT_0 ip protocol tcp ip saddr 100.64.0.11 counter snat to 198.18.0.0:23200-25215
add rule ip nat CGNATOUT_0 ip protocol udp ip saddr 100.64.0.11 counter snat to 198.18.0.0:23200-25215
```

198.18.0.0 porta origem 6767

Monitorando tráfego:

```
# apt install bmon
# bmon -b -p bond0,bond1
```

B.P.F
Brasil Peering Fórum

CGNAT Determinístico usando Mikrotik RouterOS

B.P.F
Brasil Peering Fórum

Utilizando Mikrotik RouterOS 6.x como caixa CGNAT:

- Opção **CCR1036-8G-2S+** (usando as 2 portas 10Gbps ópticas).
- Opção para caixa CGNAT Determinístico com bom custo x benefício x facilidade.
- Bons resultados com o mínimo de regras de filtro de pacotes e **Fasttrack** habilitado. Já alcancei **13 Gbps de tráfego** ou **26 Gbps agregado.**

Configurando o sistema:

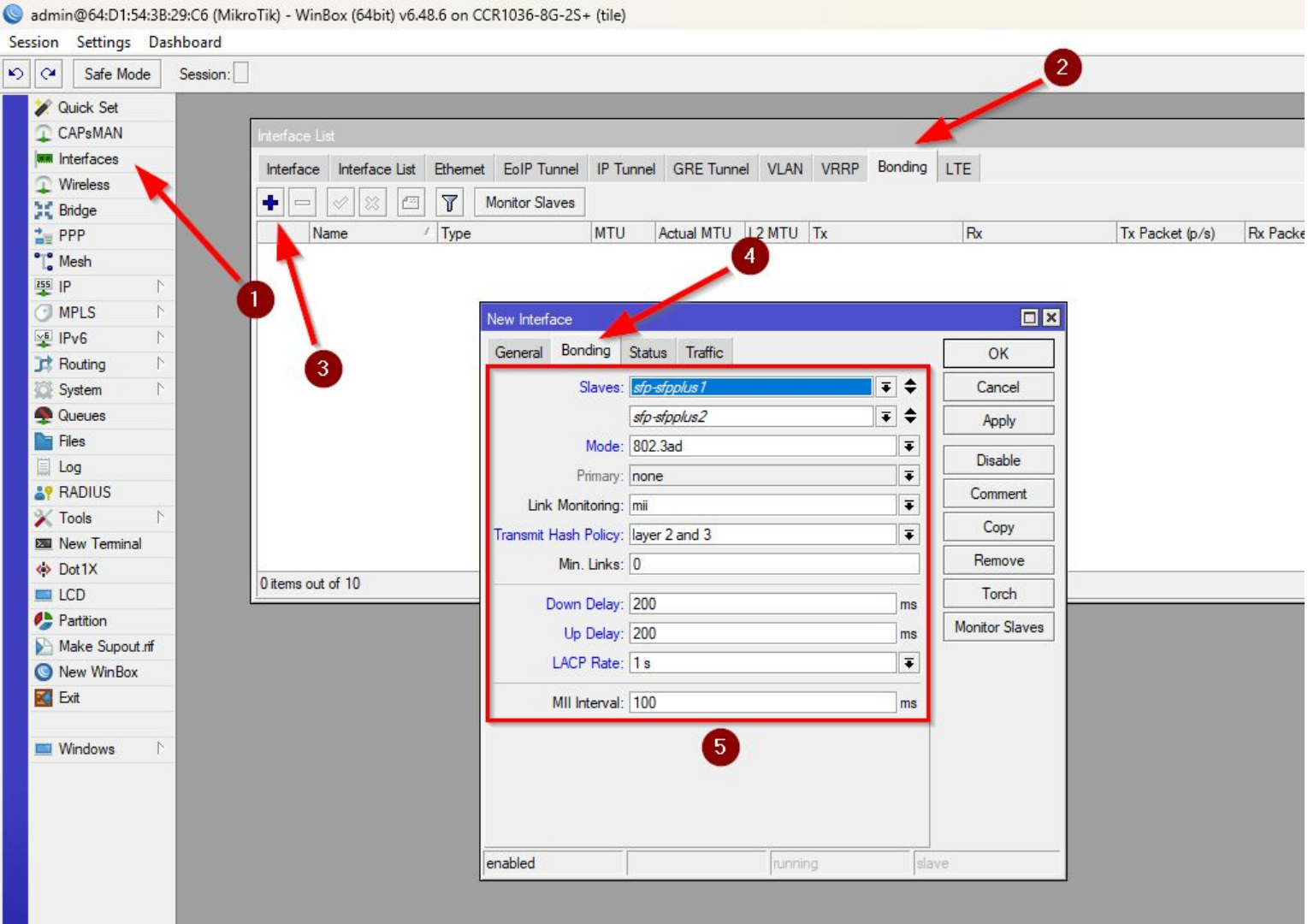
- versão **6.48.6 Long-term**, que até o momento, é a versão considerada mais estável.

- Menor complexidade de configuração com **Mikrotik RouterOS** contudo a capacidade alcançada no **Debian GNU/Linux** será superior dependendo do hardware utilizado.

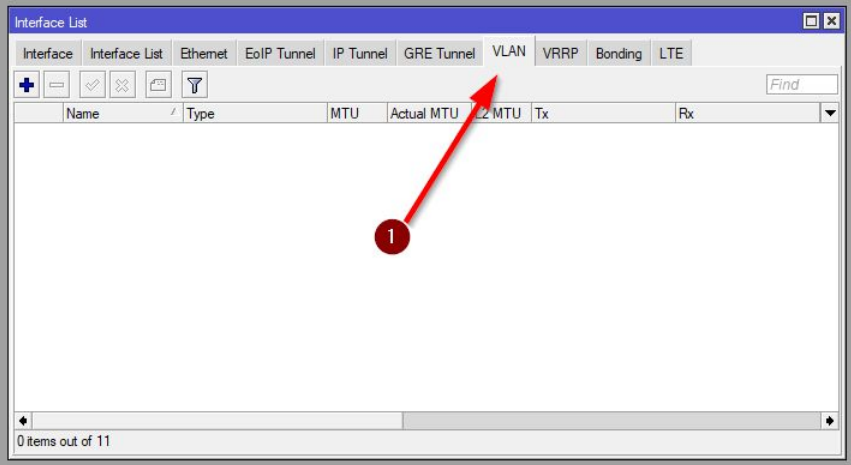
O **Fasttrack** é um recurso muito importante que aumentará a performance da sua caixa CGNAT, acelerando o encaminhamento de pacotes e diminuindo o consumo de CPU.

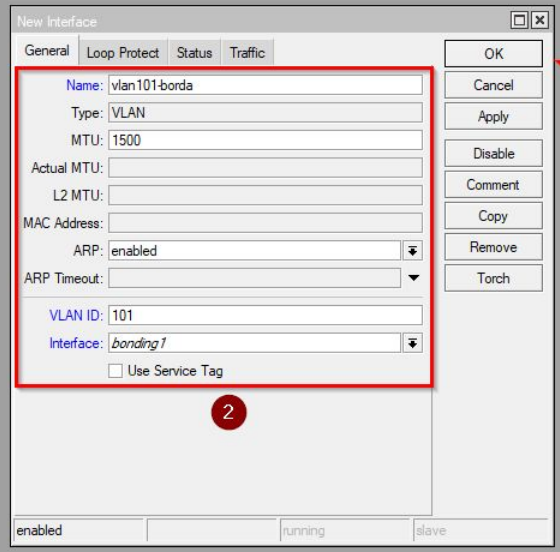
Configurando o bonding:

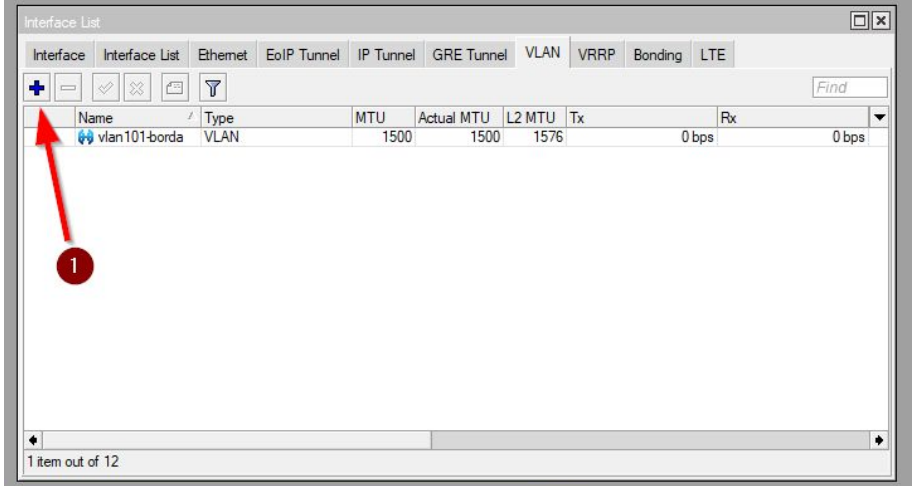
- LACP + VLAN nas portas de **10GbE sfp+** da CCR.
  - **vlan 101** interface que se comunicará com a borda.
  - **vlan 102** interface que se comunicará com o BNG.

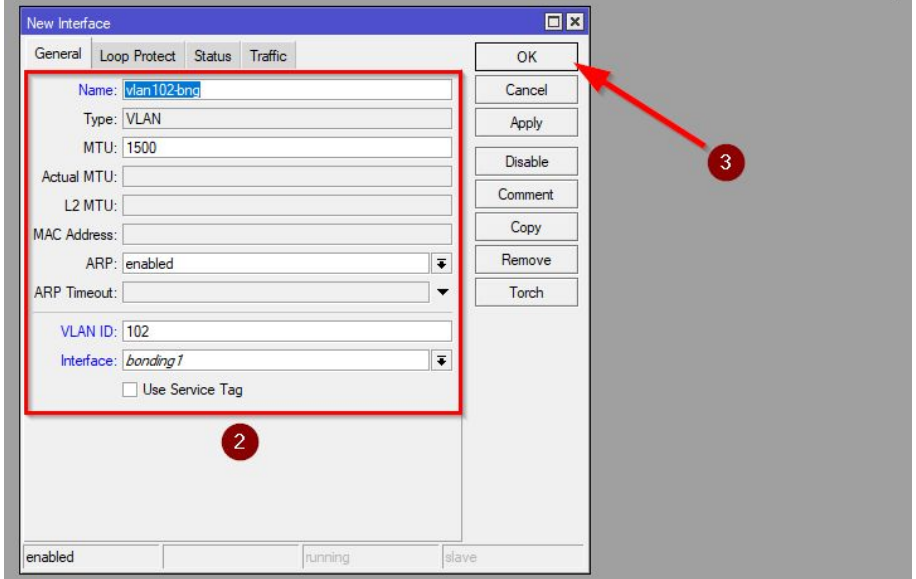


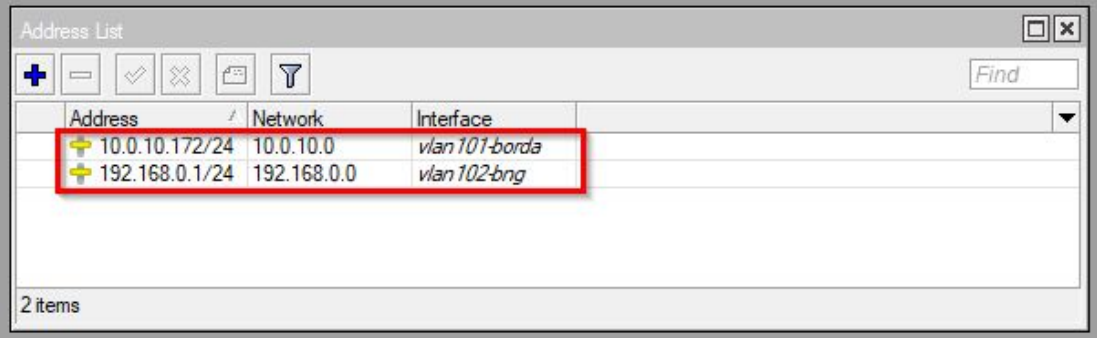
B.P.F
Brasil Peering Fórum

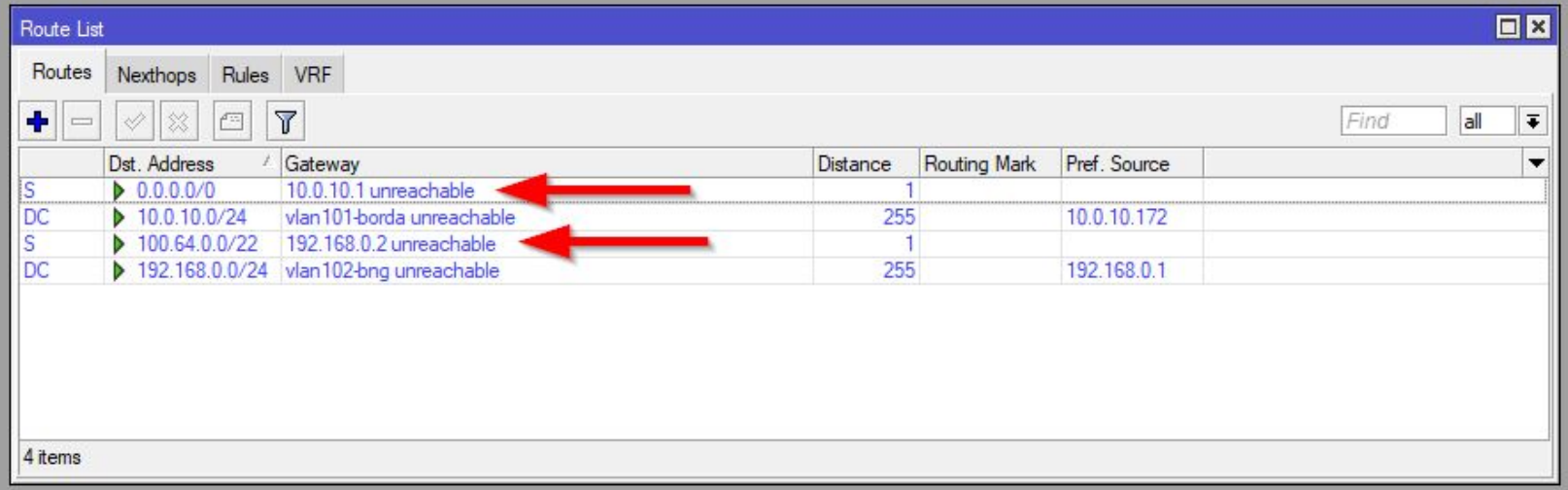

Interface List
Interface
Interface List
Ethernet
EoIP Tunnel
IP Tunnel
GRE Tunnel
VLAN
VRRP
Bonding
LTE
Find
Name
Type
MTU
Actual MTU
L2 MTU
Tx
Rx
0 items out of 11
1


New Interface
General
Loop Protect
Status
Traffic
Name: vlan101-borda
Type: VLAN
MTU: 1500
Actual MTU:
L2 MTU:
MAC Address:
ARP: enabled
ARP Timeout:
VLAN ID: 101
Interface: bonding1
Use Service Tag
OK
Cancel
Apply
Disable
Comment
Copy
Remove
Torch
2
3
enabled
running
slave


Interface List
Interface
Interface List
Ethernet
EoIP Tunnel
IP Tunnel
GRE Tunnel
VLAN
VRRP
Bonding
LTE
Find
Name
Type
MTU
Actual MTU
L2 MTU
Tx
Rx
vlan101-borda
VLAN
1500
1500
1576
0 bps
0 bps
1 item out of 12
1


New Interface
General
Loop Protect
Status
Traffic
Name: vlan102-bng
Type: VLAN
MTU: 1500
Actual MTU:
L2 MTU:
MAC Address:
ARP: enabled
ARP Timeout:
VLAN ID: 102
Interface: bonding1
Use Service Tag
OK
Cancel
Apply
Disable
Comment
Copy
Remove
Torch
2
3
enabled
running
slave





B.P.F
Brasil Peering Fórum

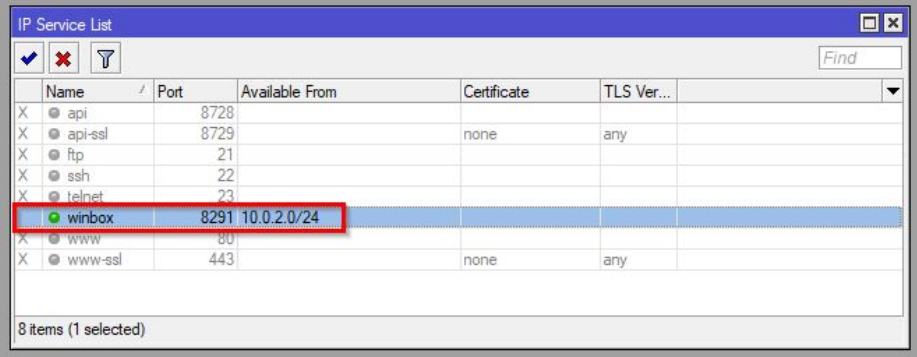
Recomendações de segurança:

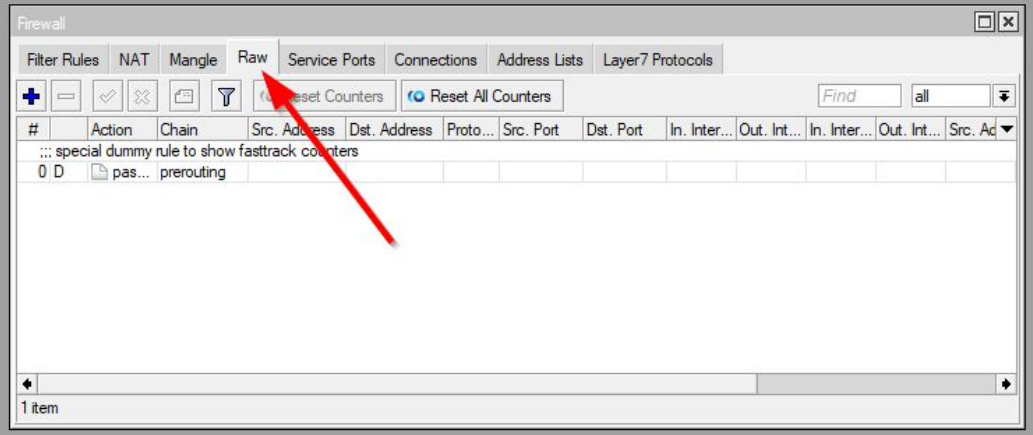
- Utilize credenciais de acesso com senhas fortes, não esqueça o **login admin sem senha** (padrão no Mikrotik RouterOS).

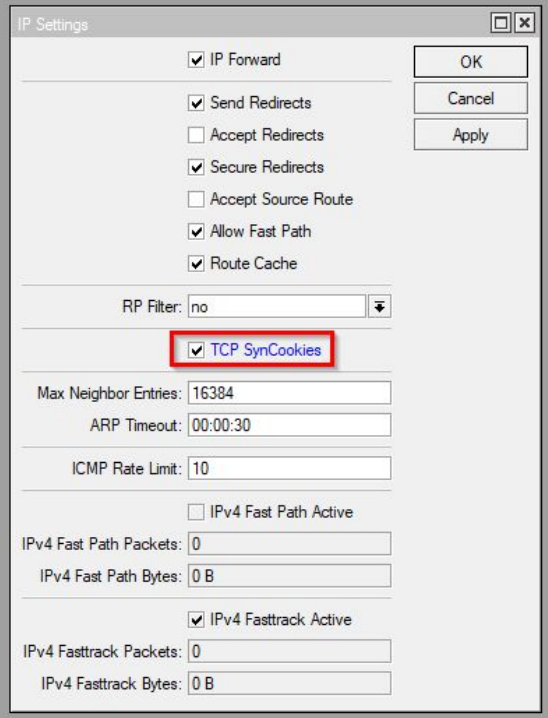
- Desabilite todos os serviços que não for utilizar e os que ficarem abertos, especifique neles o acesso apenas da sua rede de gerência. Não deixe qualquer serviço aberto para a Internet.

- Habilite o **TCP SynCookies**.

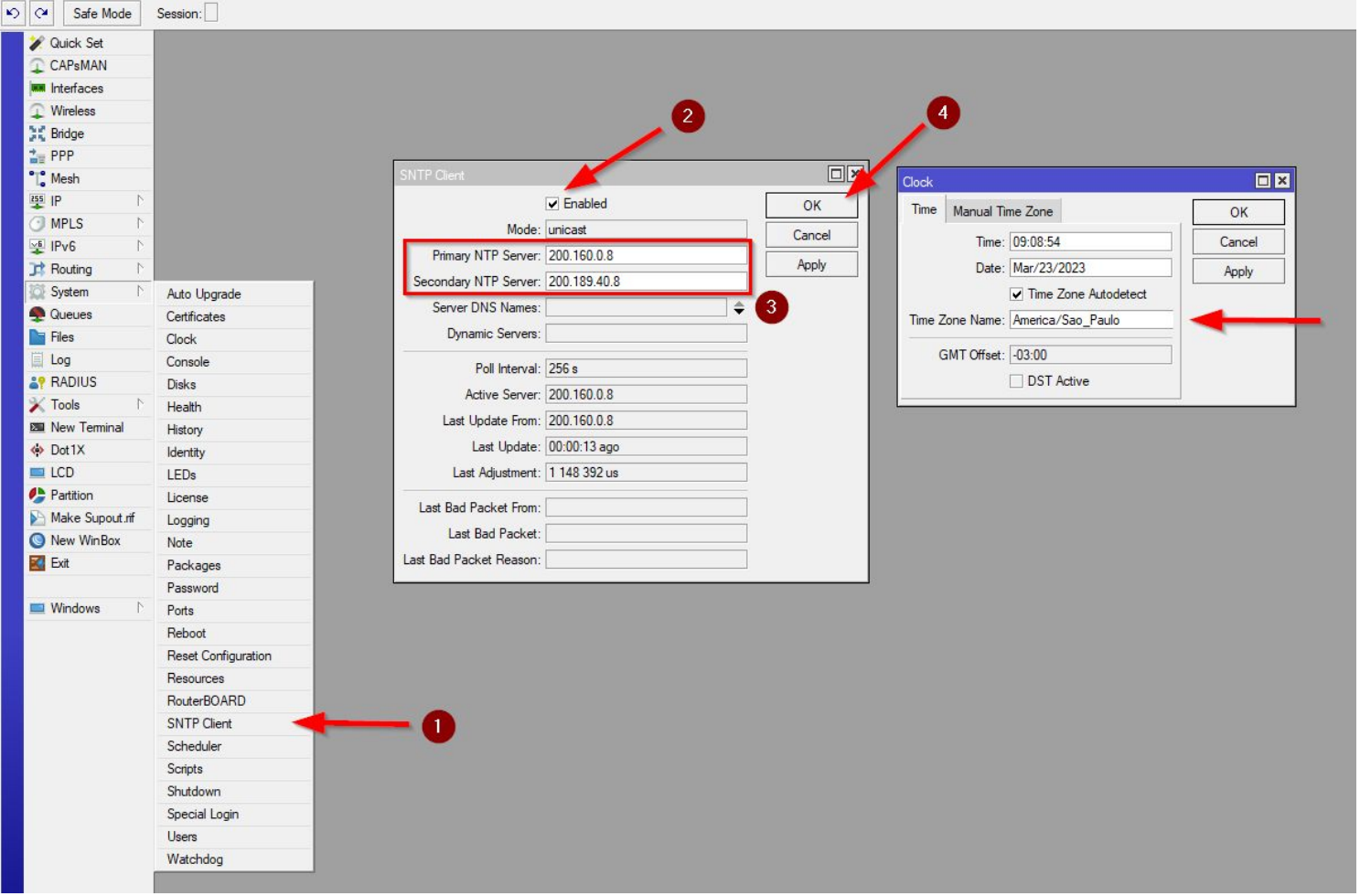
- Procure criar suas regras de filtros de pacotes sempre na **Table Raw**, ela não agride tanto a performance do equipamento mas necessita de muita atenção porque ela pode afetar os acessos dos assinantes. Isso porque uma regra genérica demais será analisada tanto com destino a caixa, destino ao cliente e o mesmo pode ocorrer no sentido inve cliente para a Internet.







B.P.F
Brasil Peering Fórum



Configure o NTP client da caixa e mantenha a data e horário adequados.

B.P.F
Brasil Peering Fórum

Criando as regras de CGNAT:

- Gerador de regras CGNAT Determinístico do **Rudimar Remontti (netmap)**.

  https://cgnat.remontti.com.br/

- **Blackholes** (**bloqueio de static loops**).
- **Fasttrack**.
- **Tabela de relacionamento**: IP público x Porta origem e IP privado.

Gerador de CGNAT para RouterOS

Networks inicial privado

100.64.0.0

Prefixo público (Network)

198.18.0.0/27

1 Público para quantos Privados?

32 Clientes [~2000 Portas]

Sequencial Chain(s)

0

Lembram dos índices que fizemos no Debian?

Ignorar destino(s)

Nenhum

Ignorar Prefixo/Lista

LISTA_SERVIDORES

No Track (RAW)

Sim

Interface Uplink

Nome da Interface (Recomendado)

Nome/Lista da interface de Uplink

vlan101-borda

Uso de Blackhole

Sim - Prefixo (Recomendado)

Nome/Lista da interface de Uplink

Loopback

Protocolos

- (•) TCP/UDP (Recomendado Judicialmente)
- ( ) Apenas TCP (Problemas Judicial*)

Fasttrack

- (•) Sim (Recomendado)
- ( ) Não

RouterOS

- (•) Versão 6.x
- ( ) Versão 7.x

Gerar Script

R A R
rudimar@remontti

Regras geradas:
49146

Se está feliz em ter encontrado uma solução fácil e rápida para gerar as regras CGNAT, lhe poupando horas de trabalho, por favor, considere fazer uma doação para apoiar o desenvolvimento e me ajudar a manter esta ferramenta acessível e ***gratuita para todos****.*
Rudimar Remontti

Quero ajudar

Contato

B.P.F
Brasil Peering Fórum

RANGE IPs PRIVADOS PARA CGNAT **100.64.0.0-100.64.3.255** | TOTAL DE IPS **1024** | PORTAS POR CLIENTE: **2016**

AO FINAL DAS REGRAS SE ENCONTRA O MAPEAMENTO DAS PORTAS.

*O **motivo** de ser tudo impresso na tela **não tendo opções para download** é que todos os dados preenchidos nunca serão armazenado. Ética!*

```
# BLACKHOLE
/ip route add type=blackhole dst-address=198.18.0.0/27 comment=CGNAT_BLACKHOLE
```

Nossa blackhole contra static loop

```
# FASTTRACK
/ip firewall filter add chain=forward action=fasttrack-connection connection-state=established,related
/ip firewall filter add chain=forward action=accept connection-state=established,related
```

Instruções para habilitar o Fasttrack

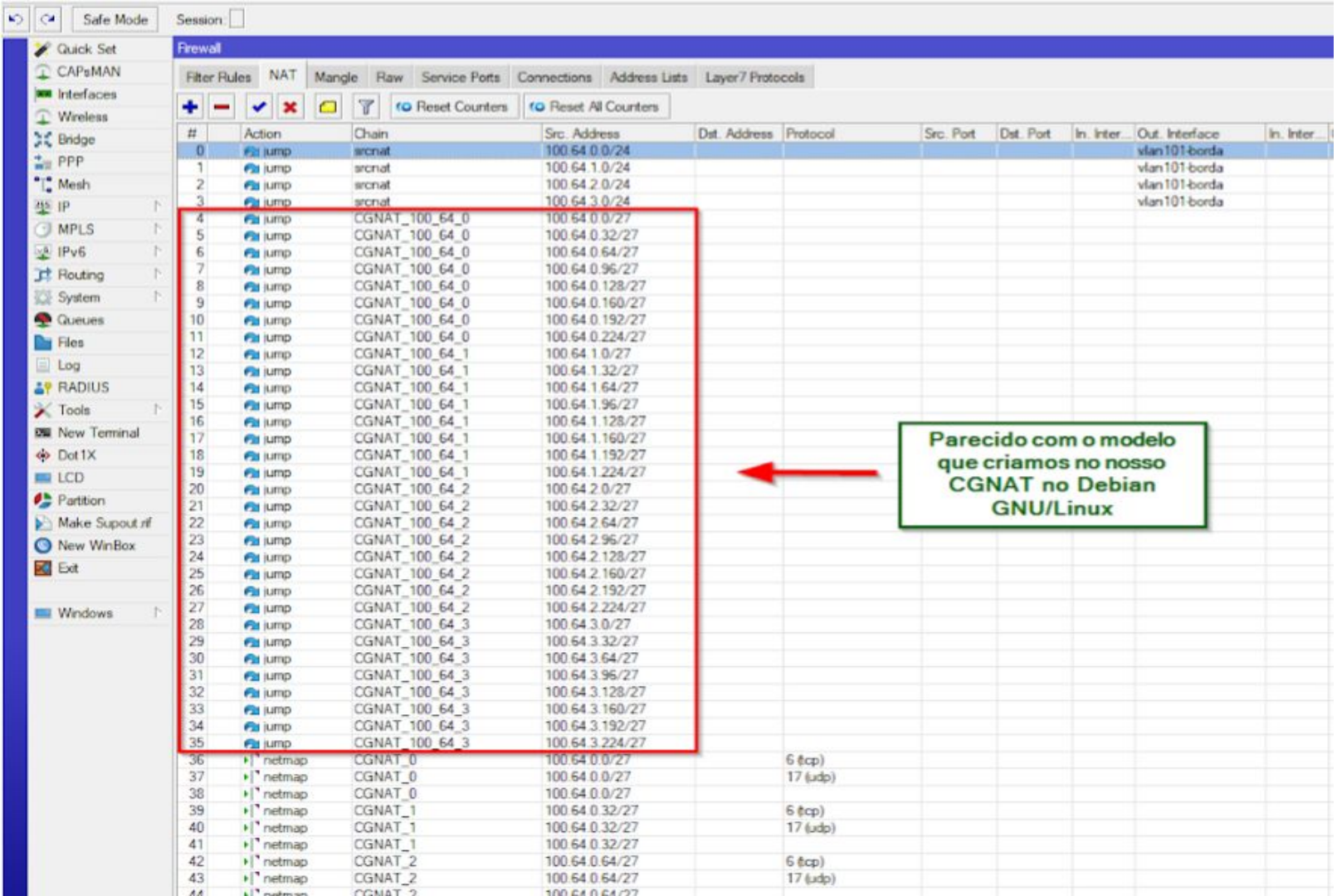
```
#CGNAT
/ip firewall nat add chain=srcnat src-address=100.64.0.0/24 out-interface="vlan101-borda"  action=jump jump-target=CGNAT_100_64_0
/ip firewall nat add chain=srcnat src-address=100.64.1.0/24 out-interface="vlan101-borda"  action=jump jump-target=CGNAT_100_64_1
/ip firewall nat add chain=srcnat src-address=100.64.2.0/24 out-interface="vlan101-borda"  action=jump jump-target=CGNAT_100_64_2
/ip firewall nat add chain=srcnat src-address=100.64.3.0/24 out-interface="vlan101-borda"  action=jump jump-target=CGNAT_100_64_3

/ip firewall nat add chain=CGNAT_100_64_0 src-address=100.64.0.0/27 action=jump jump-target=CGNAT_0
/ip firewall nat add chain=CGNAT_100_64_0 src-address=100.64.0.32/27 action=jump jump-target=CGNAT_1
/ip firewall nat add chain=CGNAT_100_64_0 src-address=100.64.0.64/27 action=jump jump-target=CGNAT_2
/ip firewall nat add chain=CGNAT_100_64_0 src-address=100.64.0.96/27 action=jump jump-target=CGNAT_3
/ip firewall nat add chain=CGNAT_100_64_0 src-address=100.64.0.128/27 action=jump jump-target=CGNAT_4
/ip firewall nat add chain=CGNAT_100_64_0 src-address=100.64.0.160/27 action=jump jump-target=CGNAT_5
/ip firewall nat add chain=CGNAT_100_64_0 src-address=100.64.0.192/27 action=jump jump-target=CGNAT_6
/ip firewall nat add chain=CGNAT_100_64_0 src-address=100.64.0.224/27 action=jump jump-target=CGNAT_7
/ip firewall nat add chain=CGNAT_100_64_1 src-address=100.64.1.0/27 action=jump jump-target=CGNAT_8
/ip firewall nat add chain=CGNAT_100_64_1 src-address=100.64.1.32/27 action=jump jump-target=CGNAT_9
/ip firewall nat add chain=CGNAT_100_64_1 src-address=100.64.1.64/27 action=jump jump-target=CGNAT_10
/ip firewall nat add chain=CGNAT_100_64_1 src-address=100.64.1.96/27 action=jump jump-target=CGNAT_11
/ip firewall nat add chain=CGNAT_100_64_1 src-address=100.64.1.128/27 action=jump jump-target=CGNAT_12
/ip firewall nat add chain=CGNAT_100_64_1 src-address=100.64.1.160/27 action=jump jump-target=CGNAT_13
/ip firewall nat add chain=CGNAT_100_64_1 src-address=100.64.1.192/27 action=jump jump-target=CGNAT_14
/ip firewall nat add chain=CGNAT_100_64_1 src-address=100.64.1.224/27 action=jump jump-target=CGNAT_15
```

Nossas regras de CGNAT propriamente ditas.

- **Tabela CGNAT de relacionamento IP privado x IP público x Porta.**

IP Público	Range de Portas	IP Privado
198.18.0.0	1024 à 3040	100.64.0.0
198.18.0.1	1024 à 3040	100.64.0.1
198.18.0.2	1024 à 3040	100.64.0.2
198.18.0.3	1024 à 3040	100.64.0.3
198.18.0.4	1024 à 3040	100.64.0.4
198.18.0.5	1024 à 3040	100.64.0.5
198.18.0.6	1024 à 3040	100.64.0.6
198.18.0.7	1024 à 3040	100.64.0.7
198.18.0.8	1024 à 3040	100.64.0.8
198.18.0.9	1024 à 3040	100.64.0.9
198.18.0.10	1024 à 3040	100.64.0.10
198.18.0.11	1024 à 3040	100.64.0.11
198.18.0.12	1024 à 3040	100.64.0.12
198.18.0.13	1024 à 3040	100.64.0.13
198.18.0.14	1024 à 3040	100.64.0.14
198.18.0.15	1024 à 3040	100.64.0.15
198.18.0.16	1024 à 3040	100.64.0.16



B.P.F
Brasil Peering Fórum

Abaixo como ficaram as regras que habilita o **Fasttrack** no nosso equipamento, aumentando em muito a performance de encaminhamento dos pacotes.

Firewall

Filter Rules | NAT | Mangle | Raw | Service Ports | Connections | Address Lists | Layer7 Protocols

Reset Counters | Reset All Counters | Find | all

#		Action	Chain	Src. Address	Dst. Address	Proto...	Src. Port	Dst. Port	In. Inter...	Out. Int...	In. Inter...	Out. Int...	Src. Ad...	Dst. Ad...	Bytes	Packets
;;; special dummy rule to show fasttrack counters																
0	D	passthrough	forward												0 B	0
1		fasttrack connection	forward												0 B	0
2		accept	forward												0 B	0

3 items

Firewall

Filter Rules | NAT | Mangle | Raw | Service Ports | Connections | Address Lists | Layer7 Protocols

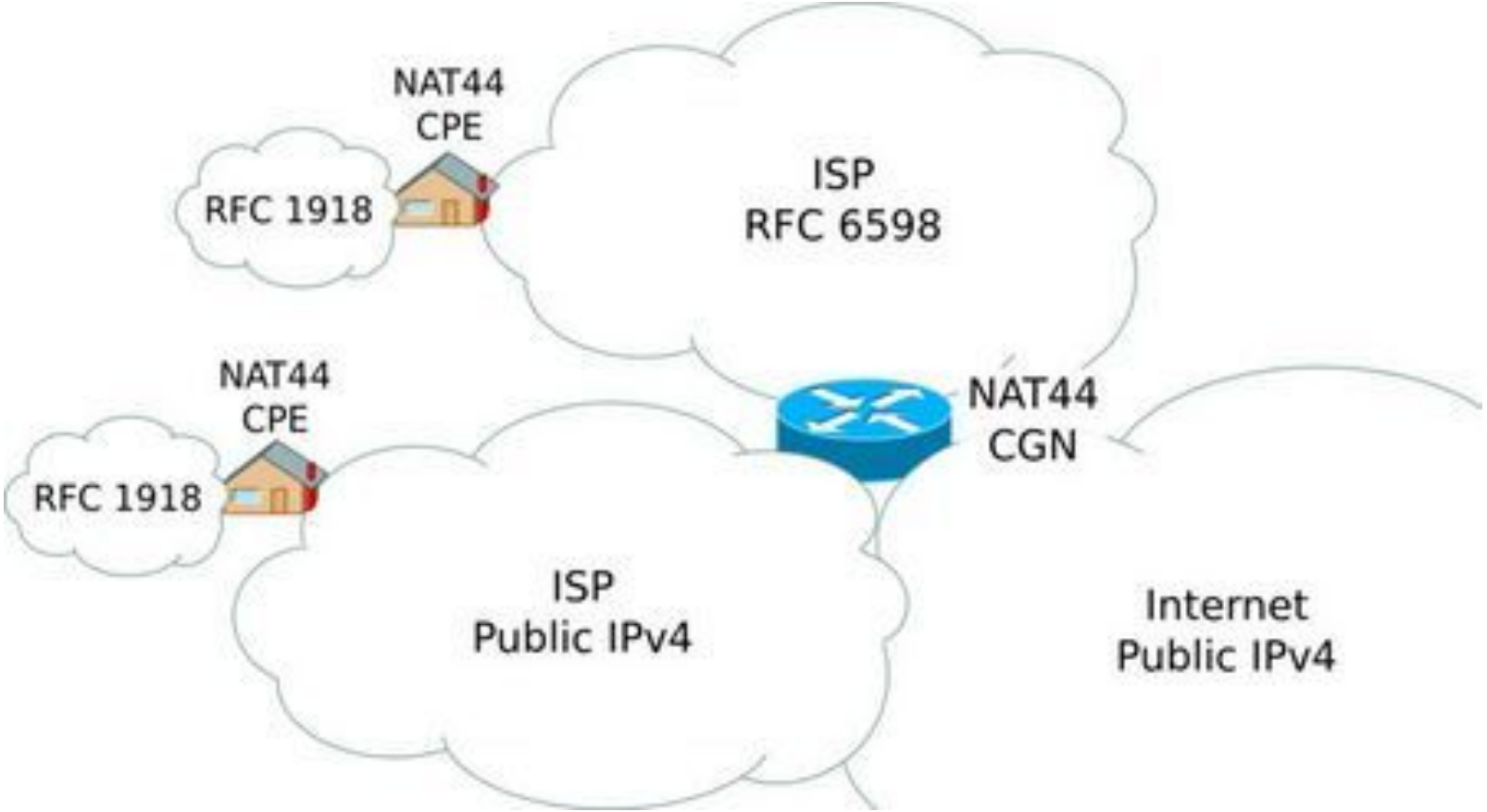
Reset Counters | Reset All Counters | Find | all

#		Action	Chain	Src. Address	Dst. Address	Proto...	Src. Port	Dst. Port	In. Inter...	Out. Int...	In. Inter...	Out. Int...	Src. Ad...	Dst. Ad...	Bytes	Packets
;;; special dummy rule to show fasttrack counters																
0	D	passthrough	prerouting												0 B	0
;;; special dummy rule to show fasttrack counters																
1	D	passthrough	forward												0 B	0
;;; special dummy rule to show fasttrack counters																
2	D	passthrough	postrouting												0 B	0

3 items

B.P.F
Brasil Peering Fórum

Servidor de Logs CGNAT SYSLOG+NETFLOW



B.P.F
Brasil Peering Fórum

Objetivo:

Armazenar logs de CGNAT para que possamos realizar as quebras de sigilo tecnológico, solicitadas nos Ofícios judiciais.

Necessidade:

- **CGNAT BPA (Bulk Port Allocation)**.
- Armazenar log do IPv6 (Mikrotik RouterOS 6.x).
- Guarda dos logs por 1 ano.

Mito:

- CGNAT BPA gera logs gigantescos.

Sugestão:

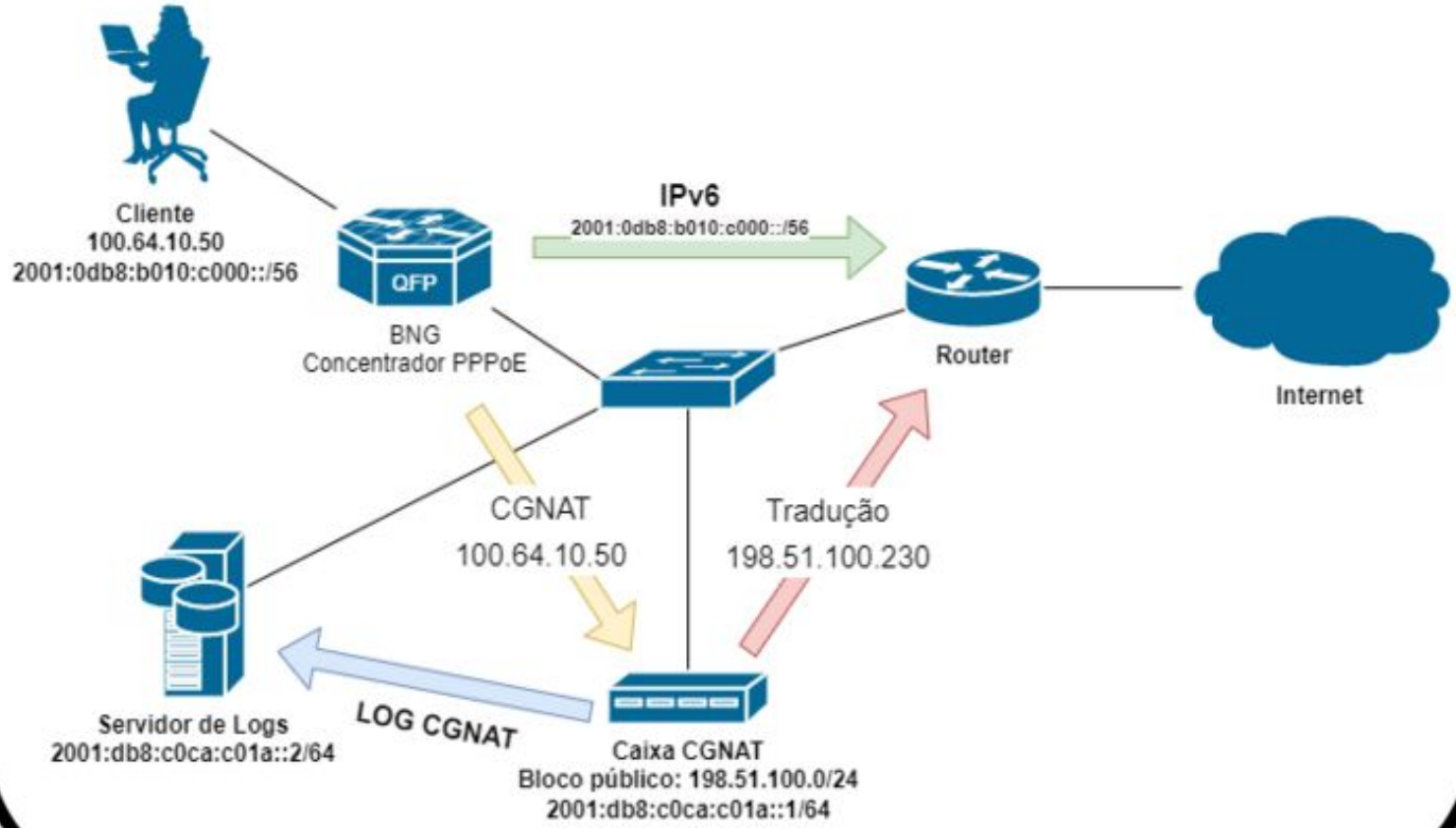
- Blocos de **256 portas**, com até **16 blocos** dando um total de até **4096 portas** por cliente.

Vantagens:

- Melhor uso dos recursos IPv4.
- Guarda de logs diários bem pequenos.

Formatos de envio dos logs:

- Syslog.
- Netflow.



B.P.F
Brasil Peering Fórum

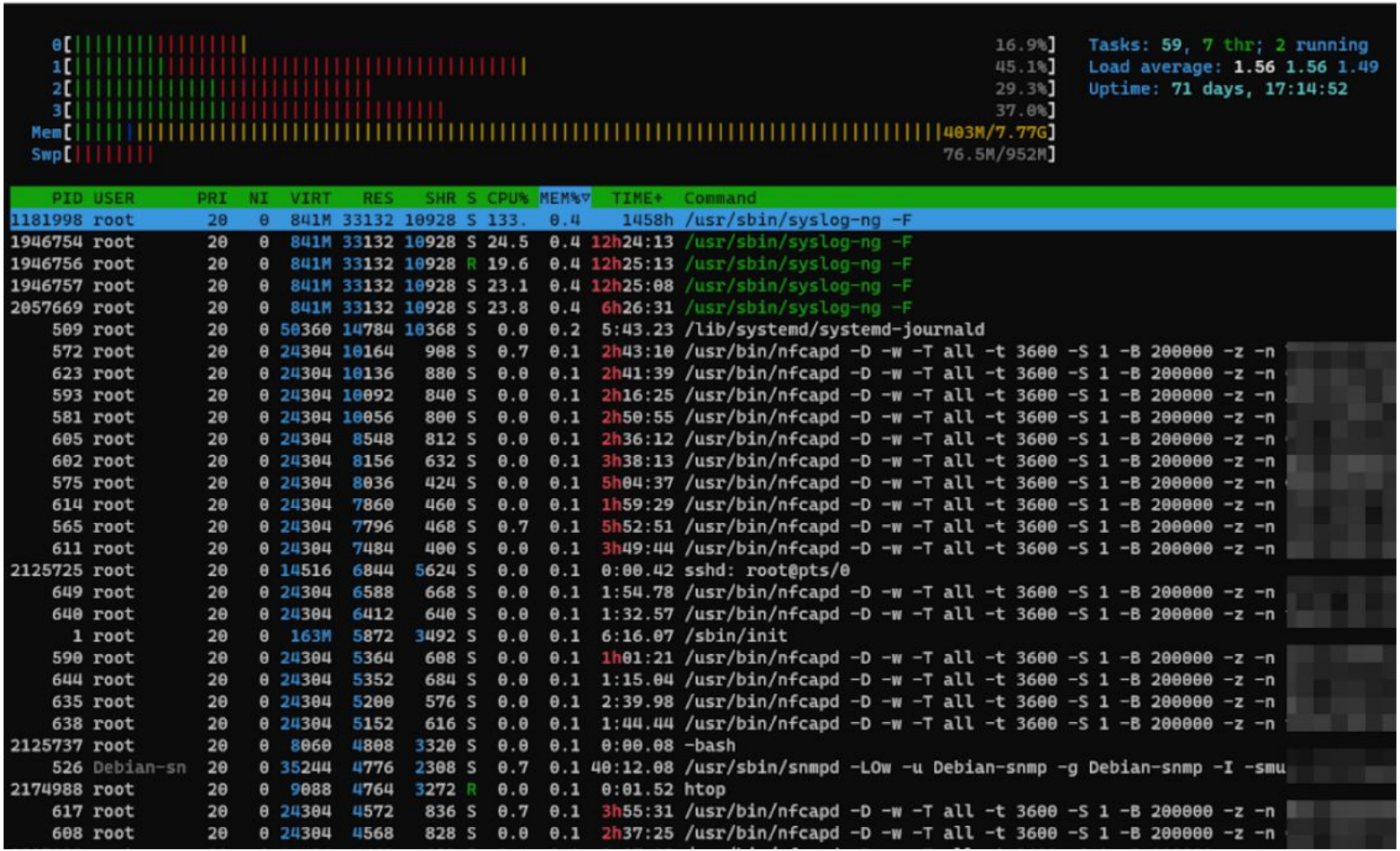
Requisitos para o sistema:

- Debian GNU/Linux 11 (Bullseye) amd64 com LVM (Logical Volume Manager).
- Nfdump (Netflow).
- Syslog-ng.
- Pigz (compactação dos logs).

Hardware:

- Espaço para armazenar seus logs com possibilidade de crescimento.
- Discos rápidos e adequados ao uso diário com I/O constante 24x7.
- Pouco processamento quando usando apenas Netflow.
- Muito pouco uso de memória.
- Segurança de disponibilidade dos dados.

- **htop** - sistema em produção.



Quanto à armazenagem desse sistema:

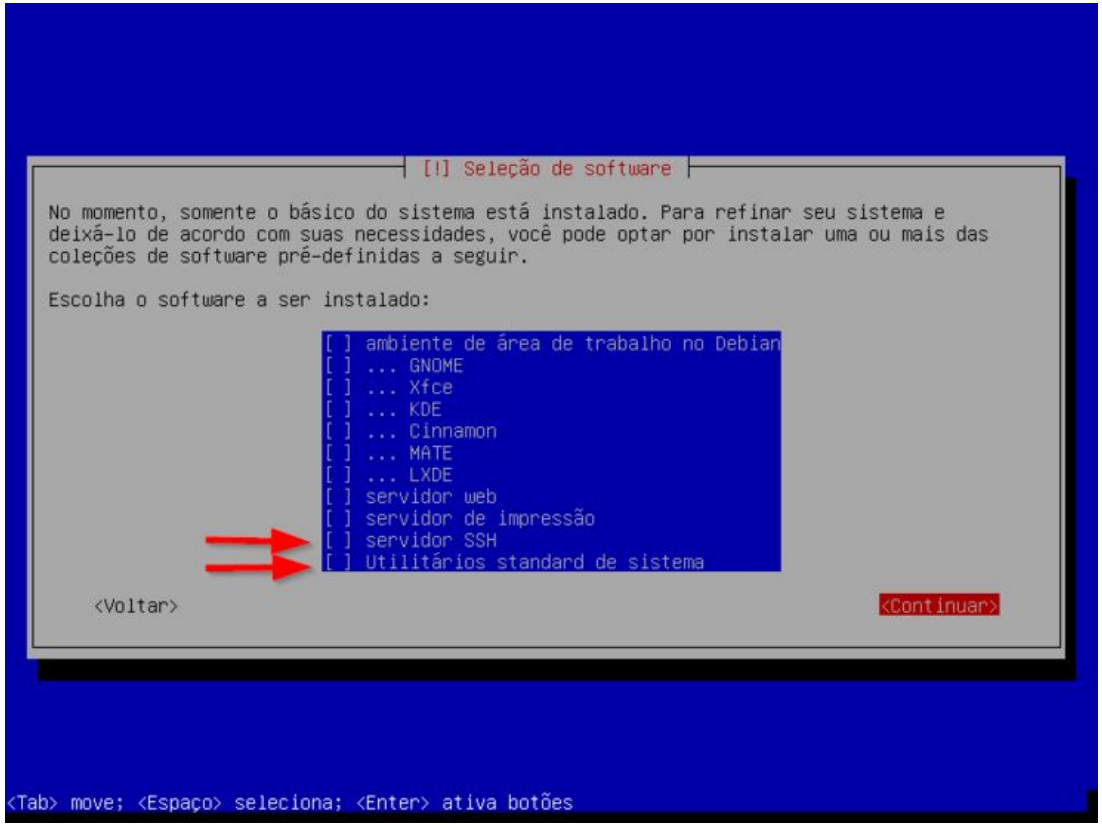
```
└──── # df -h
Sist. Arq.              Tam. Usado Disp. Uso% Montado em
udev                    3,9G     0  3,9G   0% /dev
tmpfs                   796M   81M  716M  11% /run
/dev/mapper/root-root    46G  1,5G   42G   4% /
tmpfs                   3,9G  504K  3,9G   1% /dev/shm
tmpfs                   5,0M     0  5,0M   0% /run/lock
/dev/mapper/boot-boot   451M   86M  338M  21% /boot
/dev/mapper/var-data     19T   13T  4,8T  73% /var
tmpfs                    38M     0   38M   0% /run/user/0
```

- Menos de 1 ano de logs de CGNAT armazenados.
- Uso de **LVM** para expansão.
- Sistema recebendo logs de mais de 200k assinantes.

CPU	Memória	Disco
2.4Ghz 4 cores	8G DDR4	19Tb

Preparando o Servidor:

- Configure o LVM e instale um sistema clean.



- **/etc/apt/sources.list**

```
deb http://security.debian.org/debian-security bullseye-security main contrib non-free
deb http://deb.debian.org/debian bullseye main non-free contrib
deb http://deb.debian.org/debian bullseye-updates main contrib non-free
deb http://deb.debian.org/debian bullseye-backports main contrib non-free
```

- **Algumas ferramentas e programas que usaremos:**

```
# apt install net-tools htop iotop sipcalc tcpdump curl gnupg rsync wget host dnsutils mtr-tiny bmon sudo tmux whois syslog-ng nfdump pigz chrony irqbalance
```

```
# systemctl enable irqbalance
# echo "vm.swappiness=10" >> /etc/sysctl.conf
# sysctl -p
```

NTP/NTS:

- Timezone UTC.
- Registro dos logs na data e horário do servidor de logs.

No **/etc/chrony/chrony.conf**:

```
#pool 2.debian.pool.ntp.org iburst
server a.st1.ntp.br iburst nts
server b.st1.ntp.br iburst nts
server c.st1.ntp.br iburst nts
server d.st1.ntp.br iburst nts
```

Salve o arquivo e reinicie o serviço **chronyd**:

```
# systemctl restart chronyd.service
```

Agora vamos configurar o tzdata para horário **UTC**:

```
# timedatectl set-timezone UTC
```

Formato de gravação dos logs:

/var/log/cgnat/syslog/**<HOSTNAME>**/**<ANO>**/**<MÊS>**/**<DIA>**/server-**<HORA>**.log

```
# mkdir -p /var/log/cgnat/syslog
```

- **<HOSTNAME>** é o nome do host, recebido do equipamento que estamos coletando nossos dados.
- **<ANO>**, **<MÊS>**, **<DIA>** e **<HORA>** serão usados pelo **syslog-ng** para automaticamente criar essa estrutura de diretórios e arquivos de forma organizada.

```
┌─[root@[redacted]]─[/var/log/cgnat/syslog/[redacted]CE01/2022/10/24]─[14:50:57]
└── # l
total 406100
-rw-r----- 1 root adm 22546030 out 24 00:59 server-00.log.gz
-rw-r----- 1 root adm 22295409 out 24 01:59 server-01.log.gz
-rw-r----- 1 root adm 19511381 out 24 02:59 server-02.log.gz
-rw-r----- 1 root adm 15104138 out 24 03:59 server-03.log.gz
-rw-r----- 1 root adm 12159922 out 24 04:59 server-04.log.gz
-rw-r----- 1 root adm 10203229 out 24 05:59 server-05.log.gz
-rw-r----- 1 root adm  8778175 out 24 06:59 server-06.log.gz
-rw-r----- 1 root adm  8152663 out 24 07:59 server-07.log.gz
-rw-r----- 1 root adm  9560271 out 24 08:59 server-08.log.gz
-rw-r----- 1 root adm 11836170 out 24 09:59 server-09.log.gz
-rw-r----- 1 root adm 14059478 out 24 10:59 server-10.log.gz
-rw-r----- 1 root adm 14372733 out 24 11:59 server-11.log.gz
-rw-r----- 1 root adm 15007419 out 24 12:59 server-12.log.gz
-rw-r----- 1 root adm 16748918 out 24 13:59 server-13.log.gz
-rw-r----- 1 root adm 20060654 out 24 14:59 server-14.log.gz
-rw-r----- 1 root adm 23116700 out 24 15:59 server-15.log.gz
-rw-r----- 1 root adm 23987038 out 24 16:59 server-16.log.gz
-rw-r----- 1 root adm 22258049 out 24 17:59 server-17.log.gz
-rw-r----- 1 root adm 21300876 out 24 18:59 server-18.log.gz
-rw-r----- 1 root adm 20743376 out 24 19:59 server-19.log.gz
-rw-r----- 1 root adm 17252096 out 24 20:59 server-20.log.gz
-rw-r----- 1 root adm 19978342 out 24 21:59 server-21.log.gz
-rw-r----- 1 root adm 22689108 out 24 22:59 server-22.log.gz
-rw-r----- 1 root adm 24072178 out 24 23:59 server-23.log.gz
┌─[root@[redacted]]─[/var/log/cgnat/syslog/[redacted]CE01/2022/10/24]─[14:51:00]
└── #
```

B.P.F
Brasil Peering Fórum

Observação: o **/var/log/cgnat** precisa estar dentro do volume **LVM**.

- /etc/syslog-ng/syslog-ng.conf:

```
options { chain_hostnames(off); flush_lines(0); use_dns(no); use_fqdn(no);
keep_hostname (yes); dns_cache(no); owner("root"); group("adm");
perm(0640); dir_perm(0700); create_dirs (yes); stats_freq(0);
bad_hostname("^gconfd$"); keep-timestamp(off);
};
```

- /etc/syslog-ng/conf.d/isp.conf:

```
source s_net {
   udp6(ip("2001:db8:c0ca:c01a::2") port(514));
};

destination d_ce {
file("/var/log/cgnat/syslog/${HOST}/${YEAR}/${MONTH}/${DAY}/server-${HOUR}.log"); };

filter f_ce { facility(daemon) and not message(".*SSH.*"); };
filter f_ce_ipv6 { facility(syslog); };

log { source(s_net); filter(f_ce); destination(d_ce); };
log { source(s_net); filter(f_ce_ipv6); destination(d_ce); };
```

```
# systemctl restart syslog-ng.service
```

Compactando os logs de syslog diariamente:

```
# mkdir -p /root/scripts
```

/root/scripts/compacta_syslog.sh:

```
#!/bin/bash

ANO=$(date -d "-1 day" '+%Y')
MES=$(date -d "-1 day" '+%m')
DIA=$(date -d "-1 day" '+%d')
for lista in /var/log/cgnat/syslog/*; do
  if [ -d $lista/$ANO/$MES/$DIA ]; then
      pigz -p4 --fast $lista/$ANO/$MES/$DIA/*
  fi
done
```

Programando a execução diária:

```
# chmod 700 /root/scripts/compacta_syslog.sh
# echo "00 4    * * *   root    /root/scripts/compacta_syslog.sh" >>
/etc/crontab
```

Configurando o Netflow:

- **nfcapd** do pacote **nfdump**.
- **Configurar Netflow** versão 9 nos equipamentos.

```
/usr/bin/nfcapd -D -w -T all -t 3600 -S 1 -B 200000 -z -n
RJO-DC01-CGNAT-01,2001:db8:c0ca:c01a::1,/var/log/cgnat/flow/RJO-DC
01-CGNAT-01 -b 2001:db8:c0ca:c01a::2 -p 2055
```

- **“-D”** (daemon).
- **“-w”** e **“-t”** tempo de rotacionamento **3600s**.
- **“-T all”** habilita implementações de Netflow.
- **“-S 1”** seta o formato da hierarquia de diretórios **year/month/day** automaticamente.
- **“-B 200000”** aumento de buffer.
- **“-z”** comprime o flow em **LZO1X-1**.
- **“-n”** é o **HOSTNAME**, **IP** e **diretório base**.
- **“-b 2001:db8:c0ca:c01a::2”** IP do servidor de logs CGNAT.
- **“-p 2055”** porta udp para receber o flow deste equipamento.

Criar o diretório do flow:

```
# mkdir -p /var/log/cgnat/flow/RJO-DC01-CGNAT-01
```

Criar o **/etc/rc.local**:

```
#!/bin/sh -e

/usr/bin/nfcapd -D -w -T all -t 3600 -S 1 -B 200000 -z -n
RJO-DC01-CGNAT-01,2001:db8:c0ca:c01a::1,/var/log/cgnat/flow/RJO-DC01-C
GNAT-01 -b 2001:db8:c0ca:c01a::2 -p 2055

exit 0
```

```
# chmod +x /etc/rc.local
```

B.P.F
Brasil Peering Fórum

Compactando os logs de netflow diariamente:

/root/scripts/compacta_flow.sh:

```
#!/bin/bash

ANO=$(date -d "-1 day" '+%Y')
MES=$(date -d "-1 day" '+%m')
DIA=$(date -d "-1 day" '+%d')
for FOLDER in /var/log/cgnat/flow/*; do
  if [ -d $FOLDER/$ANO/$MES/$DIA ]; then
     cd $FOLDER/$ANO/$MES/$DIA
     echo "Compactando: ${FOLDER}/$ANO/$MES/$DIA/"
     pigz -p4 --fast nfcapd*
  fi
done
```

```
# chmod 700 /root/scripts/compacta_flow.sh
# echo "00 4    * * *   root    /root/scripts/compacta_flow.sh" >> /etc/crontab
```

- **Quando o mito se torna verdade.**
- **Configuração indesejável em caixas de CGNAT BPA.**
- **Mais de 400Mb de log/hora.**

```
# cd /var/log/cgnat/flow/RJO-DC05-CGNAT01/2022/08/04
# cat nfcapd.202208042300|nfdump -r -
2022-08-04 22:59:59.992    ADD  Ignore TCP    100.64.202.178:50419 ->        0.0.0.0:0
170.XXX.XXX.XXX:50419 ->        0.0.0.0:0          0       0
2022-08-04 23:00:31.369 DELETE  Ignore TCP    100.64.202.178:50419 ->        0.0.0.0:0
170.XXX.XXX.XXX:50419 ->        0.0.0.0:0          0       0
2022-08-04 23:13:34.417    ADD  Ignore UDP    100.64.199.100:50419 ->        0.0.0.0:0
170.XXX.XXX.XXX:50419 ->        0.0.0.0:0          0       0
2022-08-04 23:15:35.908 DELETE  Ignore UDP    100.64.199.100:50419 ->        0.0.0.0:0
170.XXX.XXX.XXX:50419 ->        0.0.0.0:0          0       0
2022-08-04 23:19:48.215    ADD  Ignore TCP    100.64.199.100:50419 ->        0.0.0.0:0
170.XXX.XXX.XXX:50419 ->        0.0.0.0:0          0       0
2022-08-04 23:20:19.283 DELETE  Ignore TCP    100.64.199.100:50419 ->        0.0.0.0:0
170.XXX.XXX.XXX:50419 ->        0.0.0.0:0          0       0
2022-08-04 23:41:02.639    ADD  Ignore UDP    100.64.204.78:50419 ->        0.0.0.0:0
170.XXX.XXX.XXX:50419 ->        0.0.0.0:0          0       0
2022-08-04 23:43:02.435 DELETE  Ignore UDP    100.64.204.78:50419 ->        0.0.0.0:0
170.XXX.XXX.XXX:50419 ->        0.0.0.0:0          0       0
```

- Configuração desejável. Entrega dinâmica por bloco de portas.
- Logs menores que **500Kb/hora**.

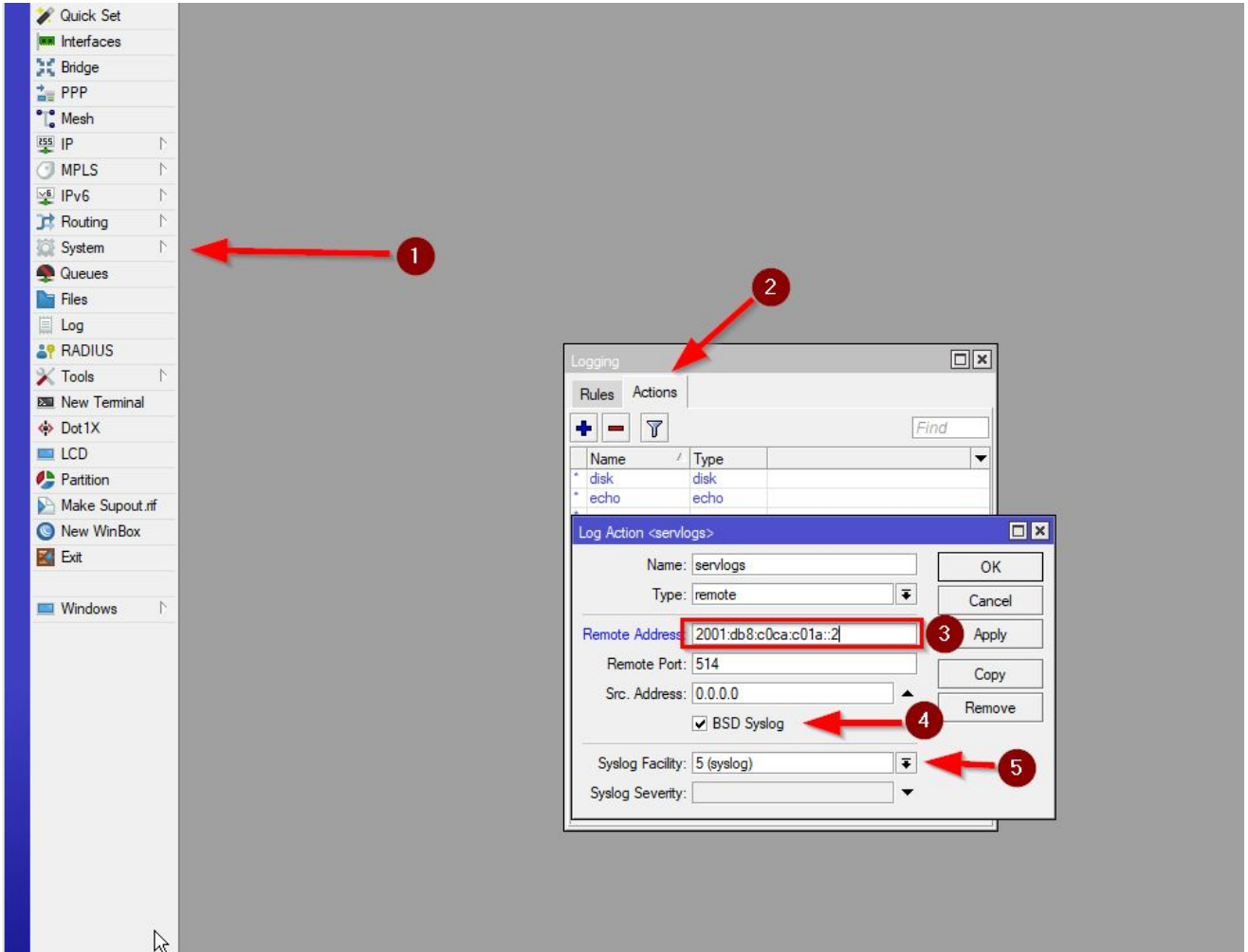
```
# cd /var/log/cgnat/flow/CGN-BOX1-SPO01/2023/01/15
# cat nfcapd.202301151100 | nfdump -o "fmt:%ts %nevt %pr %sa %nda
%nsa %pbstart %pbend" -r -
```

Date first seen	Event	Proto	Src IP Addr	X-late	Dst IP	X-late Src IP	Pb-Start	Pb-End
2023-01-15 10:54:48.746	ADD	255	100.113.127.23		0.0.0.0	186	8192	8447
2023-01-15 10:54:48.858	ADD	255	100.113.24.254		0.0.0.0	186.	5888	6143
2023-01-15 10:54:51.271	DELETE	255	100.113.13.121		0.0.0.0	170	31488	31743
2023-01-15 10:54:51.070	ADD	255	100.113.94.129		0.0.0.0	18	13568	13823
2023-01-15 10:54:52.510	ADD	255	100.113.42.63		0.0.0.0	170	22784	23039
2023-01-15 10:54:53.925	DELETE	255	100.113.124.135		0.0.0.0	186	13312	13567
2023-01-15 10:54:56.343	ADD	255	100.113.110.240		0.0.0.0	170	19200	19455
2023-01-15 10:54:57.087	ADD	255	100.113.13.11		0.0.0.0	170	18944	19199
2023-01-15 10:54:57.850	DELETE	255	100.113.73.36		0.0.0.0	186.	34048	34303
2023-01-15 10:54:58.797	DELETE	255	100.113.109.3		0.0.0.0	186.	8704	8959
2023-01-15 10:54:58.659	DELETE	255	100.113.65.240		0.0.0.0	170	46848	47103
2023-01-15 10:54:59.603	DELETE	255	100.113.37.179		0.0.0.0	170	52224	52479
2023-01-15 10:55:00.213	ADD	255	100.113.0.250		0.0.0.0	186	65280	65535
2023-01-15 10:55:00.826	DELETE	255	100.113.75.141		0.0.0.0	170	21760	22015
2023-01-15 10:55:00.662	ADD	255	100.113.66.204		0.0.0.0	170	38144	38399
2023-01-15 10:55:01.889	ADD	255	100.113.123.228		0.0.0.0	186	23808	24063
2023-01-15 10:55:02.433	DELETE	255	100.113.4.173		0.0.0.0	170	37888	38143
2023-01-15 10:55:01.579	ADD	255	100.113.42.169		0.0.0.0	186	41472	41727
2023-01-15 10:55:01.611	ADD	255	100.113.67.40		0.0.0.0	186	4864	5119
2023-01-15 10:55:03.107	ADD	255	100.113.31.84		0.0.0.0	170	35584	35839
2023-01-15 10:55:07.126	ADD	255	100.113.27.244		0.0.0.0	170	59392	59647
2023-01-15 10:55:07.274	DELETE	255	100.113.103.238		0.0.0.0	186.	39424	39679
2023-01-15 10:55:06.591	ADD	255	100.113.77.243		0.0.0.0	170	64512	64767
2023-01-15 10:55:07.907	DELETE	255	100.113.10.22		0.0.0.0	170	50176	50431
2023-01-15 10:55:08.305	DELETE	255	100.113.82.142		0.0.0.0	170	20992	21247
2023-01-15 10:55:08.050	DELETE	255	100.113.30.31		0.0.0.0	170	32512	32767
2023-01-15 10:55:09.330	ADD	255	100.113.25.52		0.0.0.0	186	48896	49151
2023-01-15 10:55:10.129	ADD	255	100.113.42.35		0.0.0.0	170	14848	15103
2023-01-15 10:55:10.545	ADD	255	100.113.37.250		0.0.0.0	170	65024	65279
2023-01-15 10:55:10.689	ADD	255	100.113.57.38		0.0.0.0	170	55808	56063
2023-01-15 10:55:12.583	ADD	255	100.113.115.48		0.0.0.0	170	13056	13311
2023-01-15 10:55:12.554	ADD	255	100.113.114.74		0.0.0.0	170	30464	30719
2023-01-15 10:55:14.067	DELETE	255	100.113.92.178		0.0.0.0	170	47872	48127
2023-01-15 10:55:15.354	ADD	255	100.113.11.47		0.0.0.0	170	63232	63487
2023-01-15 10:55:16.694	ADD	255	100.113.116.137		0.0.0.0	170	49152	49407
2023-01-15 10:55:16.694	DELETE	255	100.113.116.137		0.0.0.0	170	49152	49407
2023-01-15 10:55:16.694	ADD	255	100.113.116.137		0.0.0.0	170	20224	20479
2023-01-15 10:55:18.263	ADD	255	100.113.115.6		0.0.0.0	170	16128	16383
2023-01-15 10:55:18.866	DELETE	255	100.113.110.28		0.0.0.0	170	23040	23295
2023-01-15 10:55:19.137	DELETE	255	100.113.116.52		0.0.0.0	170	18688	18943
2023-01-15 10:55:19.593	ADD	255	100.113.78.31		0.0.0.0	186	40192	40447
2023-01-15 10:55:19.949	DELETE	255	100.113.4.123		0.0.0.0	170	63744	63999
2023-01-15 10:55:20.441	DELETE	255	100.113.37.91		0.0.0.0	18	19968	20223
2023-01-15 10:55:21.379	DELETE	255	100.113.7.32		0.0.0.0	186	37376	37631
2023-01-15 10:55:20.898	DELETE	255	100.113.116.114		0.0.0.0	170	59904	60159
2023-01-15 10:55:21.190	ADD	255	100.113.42.220		0.0.0.0	170	32768	33023
2023-01-15 10:55:21.978	ADD	255	100.113.14.187		0.0.0.0	186	27648	27903
2023-01-15 10:55:23.067	DELETE	255	100.113.29.242		0.0.0.0	170	29696	29951
2023-01-15 10:55:23.137	ADD	255	100.113.91.149		0.0.0.0	170	25088	25343

--Mais--

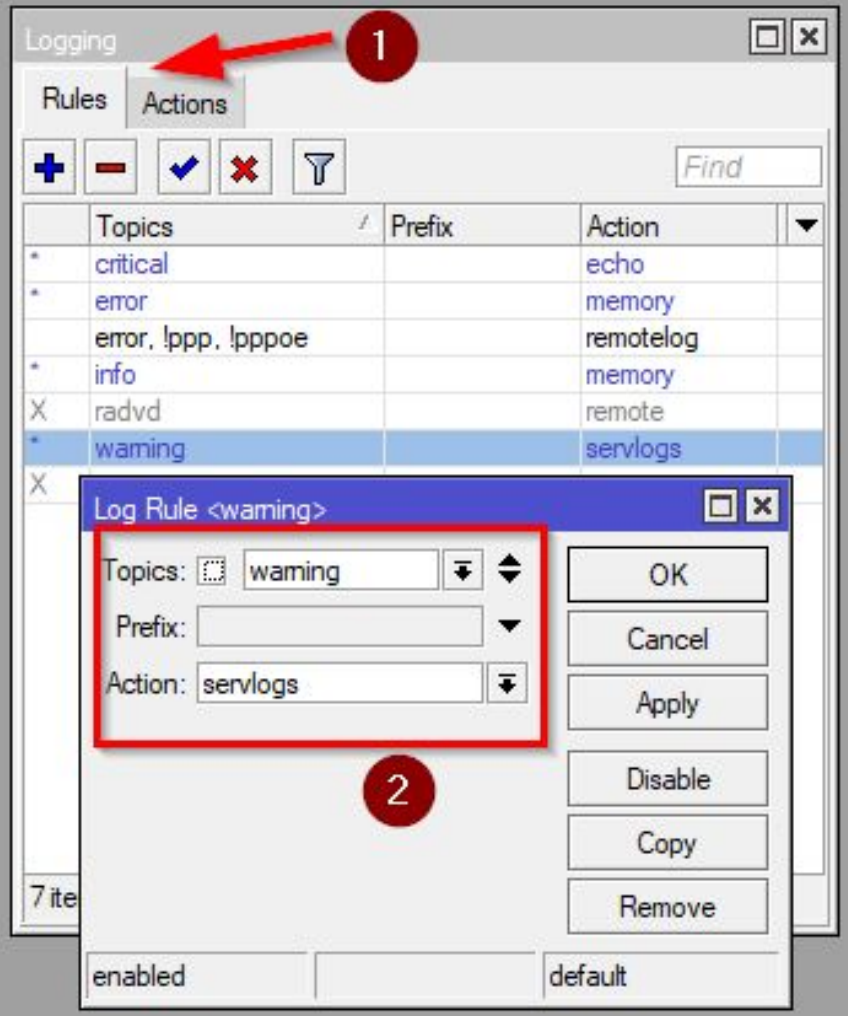
B.P.F
Brasil Peering Fórum

Configurando um Mikrotik RouterOS para envio de logs IPv6:



- Contribuição do **Bruno Viviani** neste artigo **no Brasil Peering Fórum.**

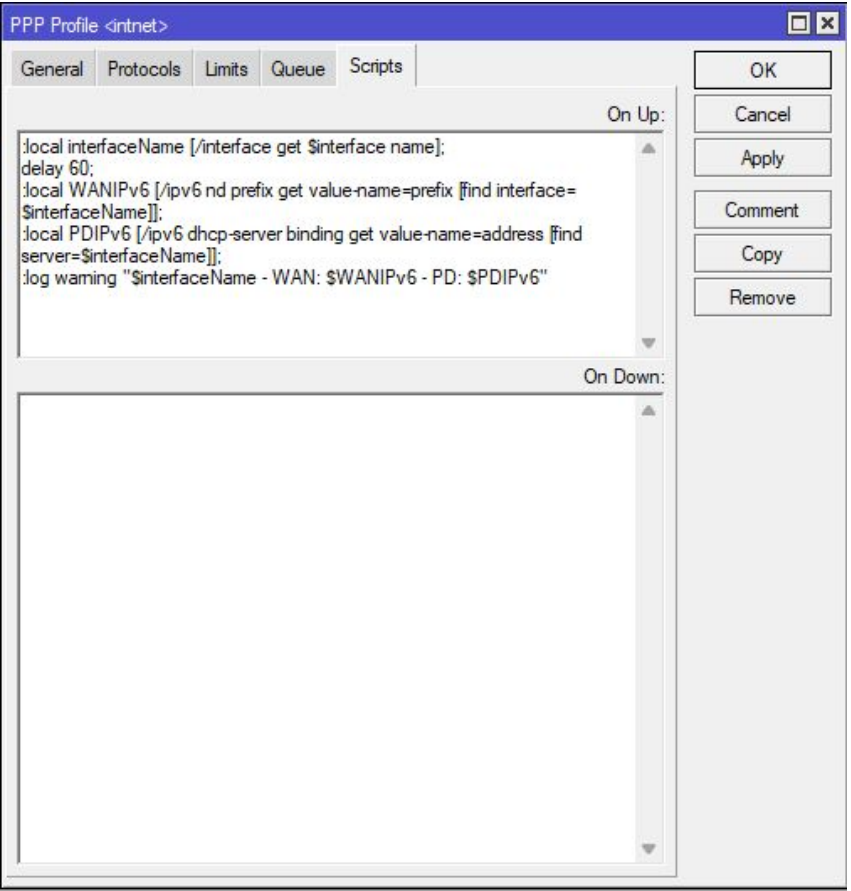
B.P.F
Brasil Peering Fórum



Modificamos a **Rule warning** e apontamos para o nosso **servlogs**:

B.P.F
Brasil Peering Fórum

```
:local interfaceName [/interface get $interface name];
delay 60;
:local WANIPv6 [/ipv6 nd prefix get value-name=prefix [find interface=$interfaceName]];
:local PDIPv6 [/ipv6 dhcp-server binding get value-name=address [find server=$interfaceName]];
:log warning "$interfaceName - WAN: $WANIPv6 - PD: $PDIPv6"
```



Por último temos que adicionar no **profile do PPPoE**, na parte de scripts em **On Up**.

B.P.F
Brasil Peering Fórum

Nos logs do servidor aparecerão algo assim:

```
Jan 25 02:26:56 SPO-SEDE-CE02 <pppoe-jose> - WAN: 2804:0db8:8005:5668::/64 - PD: 2804:0db8:8050:7b00::/56
Jan 25 02:37:08 SPO-SEDE-CE02 <pppoe-maria> - WAN: 2804:0db8:8005:5669::/64 - PD: 2804:0db8:8050:a000::/56
```