

IPv6 no FreeBSD

Mauro Paes Corrêa
XI BSDDAY - UFFRJ



Características da implementação do Ipv6 no FreeBSD

Desde o FreeBSD 4.0, o IPv6 está operacional no sistema, mas ainda não habilitado por padrão

- 1 Somente a partir da versão 9.0, ele vem habilitado por padrão
- 2 Derivação da pilha de rede do IPv6 vem do Projeto KAME, sendo uma referência para IPv6 e IPsec no sistema
- 3 Permite uso da Pilha Unificada, podendo ser configurado para aceitar conexões IPv4 em Sockets IPv6, logo faz nat4to6 e vice-versa, DESDE que a máquina tenha um IPv4.
- 4 Parâmetros para certas features, como o comportamento do IPv4 em socket IPv6, são feitas via sysctl ou setsockopt

```
192.168.1.28 - PuTTY
root@freebsd:/etc # ifconfig
em0: flags=1008843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST,LOWER_UP> metric 0 mtu
1500
    options=48525bb<RXCSUM, TXCSUM, VLAN_MTU, VLAN_HWTAGGING, JUMBO_MTU, VLAN_HWC
SUM, TSO4, LRO, WOL_MAGIC, VLAN_HWFILTER, VLAN_HWTSO, HWSTATS, MEXTPG>
    ether bc:24:11:a3:9e:89
    inet 192.168.1.28 netmask 0xffffffff broadcast 192.168.1.255
    inet6 fe80::be24:11ff:fea3:9e89%em0 prefixlen 64 scopeid 0x1
    inet6 2a0f:85c1:c30:1000::66 prefixlen 64
    media: Ethernet autoselect (1000baseT <full-duplex>)
    status: active
    nd6 options=21<PERFORMNUD,AUTO_LINKLOCAL>
lo0: flags=1008049<UP,LOOPBACK,RUNNING,MULTICAST,LOWER_UP> metric 0 mtu 16384
    options=680003<RXCSUM, TXCSUM, LINKSTATE, RXCSUM_IPV6, TXCSUM_IPV6>
    inet 127.0.0.1 netmask 0xff000000
    inet6 ::1 prefixlen 128
    inet6 fe80::1%lo0 prefixlen 64 scopeid 0x2
    groups: lo
    nd6 options=21<PERFORMNUD,AUTO_LINKLOCAL>
root@freebsd:/etc #
```



Como é a implementação do IPv6 no FreeBSD?

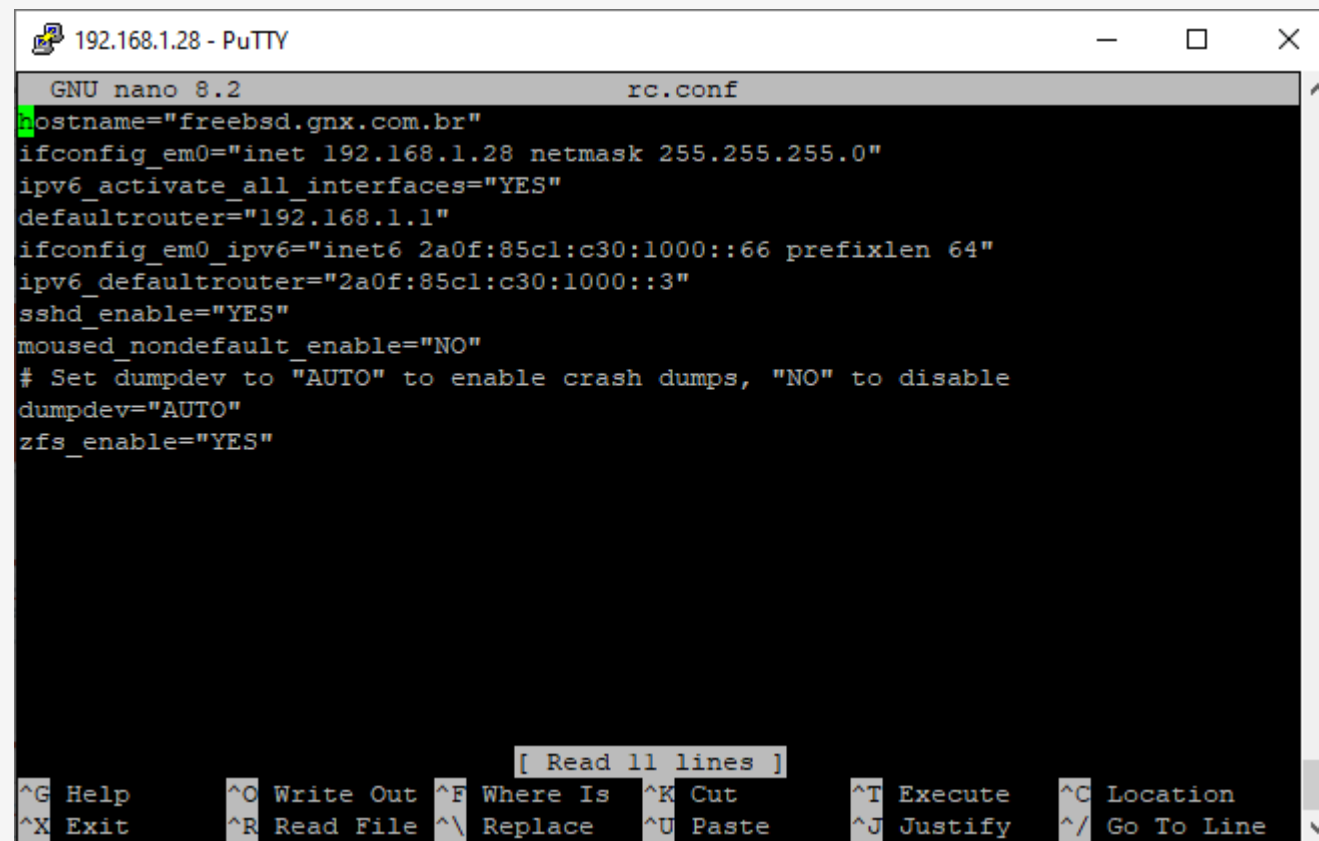
A implementação do IPv6 no FreeBSD, é fácil de ser realizada, mas o sistema precisa compreender que:

- O protocolo será ativado
- Qual endereço será utilizado, com máscara
- E qual gateway será usado

E principalmente

- **Ativar o Ipv6 no sistema!**

O parâmetro `ipv6_enable="YES"` garante a ativação da funcionalidade



The screenshot shows a PuTTY terminal window titled "192.168.1.28 - PuTTY". Inside, the GNU nano 8.2 editor is open to the file `rc.conf`. The configuration includes setting the hostname to `freebsd.gnx.com.br`, configuring the `em0` interface with IPv4 address `192.168.1.28` and netmask `255.255.255.0`, and enabling IPv6 on the same interface with address `2a0f:85c1:c30:1000::66` and prefix length `64`. It also sets the default IPv6 router to `2a0f:85c1:c30:1000::3`, enables `sshd`, disables the non-default mouse, sets crash dumps to `AUTO`, and enables `zfs`. The bottom of the window shows nano editor shortcuts.

```
GNU nano 8.2 rc.conf
hostname="freebsd.gnx.com.br"
ifconfig_em0="inet 192.168.1.28 netmask 255.255.255.0"
ipv6_activate_all_interfaces="YES"
defaultrouter="192.168.1.1"
ifconfig_em0_ipv6="inet6 2a0f:85c1:c30:1000::66 prefixlen 64"
ipv6_defaultrouter="2a0f:85c1:c30:1000::3"
sshd_enable="YES"
moused_nondefault_enable="NO"
# Set dumpdev to "AUTO" to enable crash dumps, "NO" to disable
dumpdev="AUTO"
zfs_enable="YES"
```

[Read 11 lines]

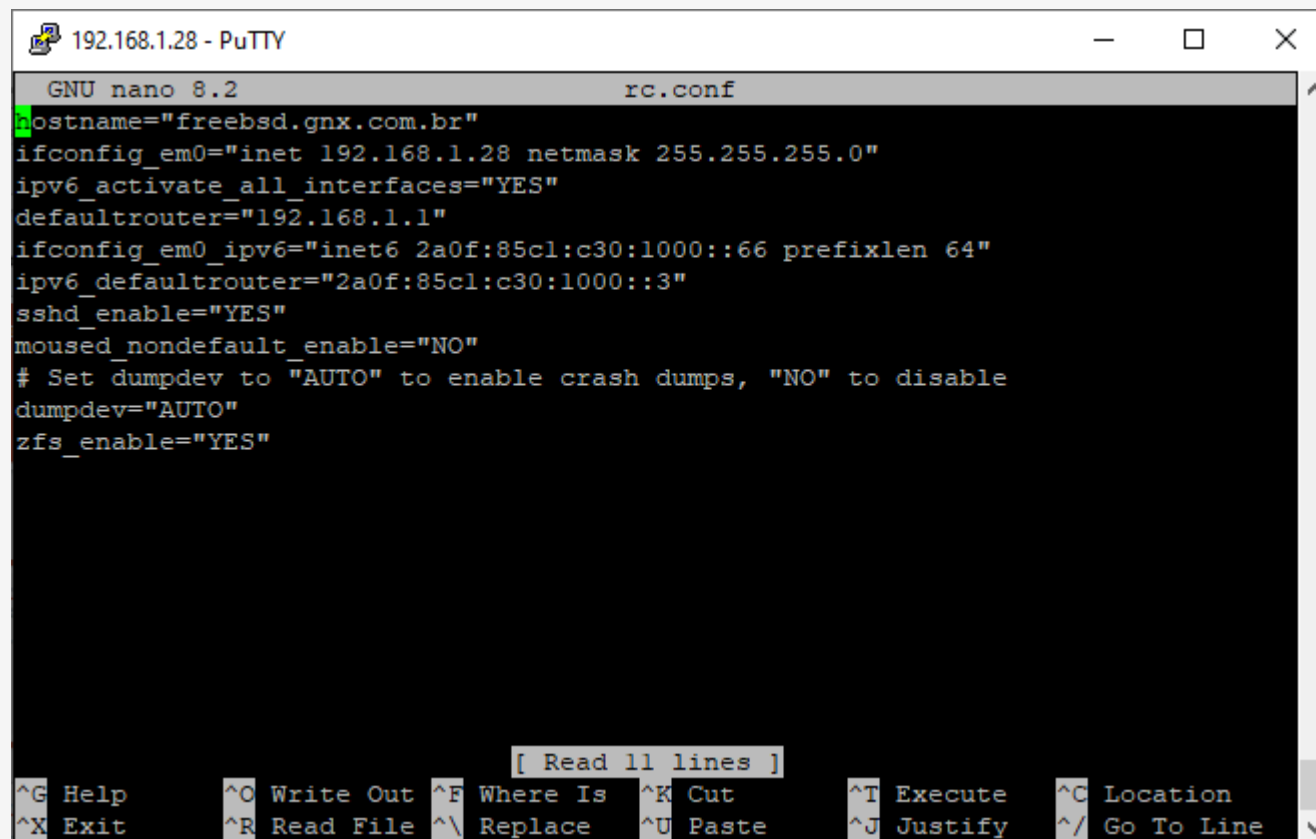
^G Help	^O Write Out	^F Where Is	^K Cut	^T Execute	^C Location
^X Exit	^R Read File	^I Replace	^U Paste	^J Justify	^_ Go To Line



Tem aliases para mais de um endereço no FreeBSD?

Sim, é possível adicionar vários endereços em uma mesma linha de parâmetro, permitindo conectividade com distintas redes, através do parâmetro:

Ifconfig_(interface)_ipv6_aliases="inet (endereço)/prefixo"



The screenshot shows a PuTTY terminal window titled "192.168.1.28 - PuTTY". Inside, the GNU nano 8.2 editor is open to the rc.conf file. The file contains the following configuration lines:

```
hostname="freebsd.gnx.com.br"
ifconfig_em0="inet 192.168.1.28 netmask 255.255.255.0"
ipv6_activate_all_interfaces="YES"
defaultrouter="192.168.1.1"
ifconfig_em0_ipv6="inet6 2a0f:85c1:c30:1000::66 prefixlen 64"
ipv6_defaultrouter="2a0f:85c1:c30:1000::3"
sshd_enable="YES"
moused_nondefault_enable="NO"
# Set dumpdev to "AUTO" to enable crash dumps, "NO" to disable
dumpdev="AUTO"
zfs_enable="YES"
```

At the bottom of the terminal, there is a status bar with the text "[Read 11 lines]" and a table of keyboard shortcuts:

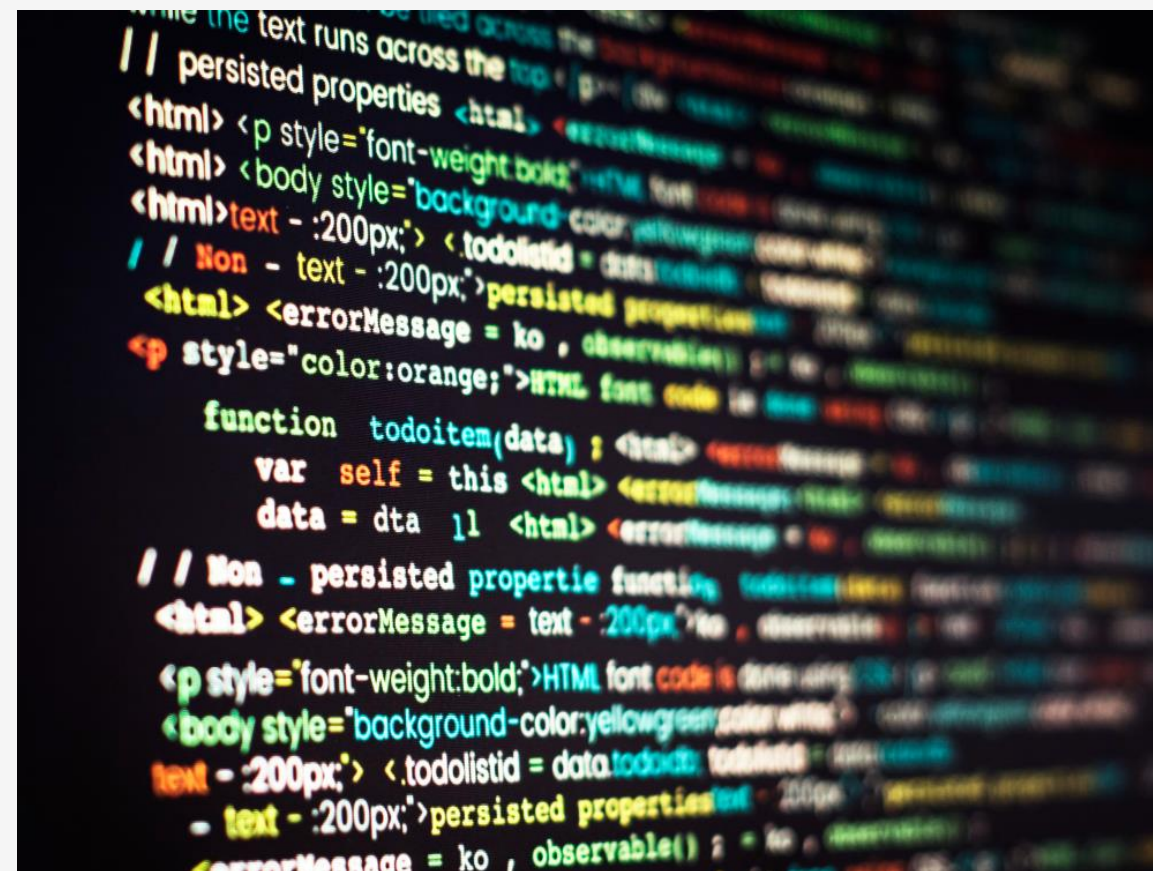
^G Help	^O Write Out	^F Where Is	^K Cut	^T Execute	^C Location
^X Exit	^R Read File	^_ Replace	^U Paste	^J Justify	^_ Go To Line



Curiosidade na prática...

Você sabia que seria totalmente possível dar para cada GRÃO de areia, um endereço IPv6?

- Número estimado de grãos de areia: 10^{23} ,
- Número de IPV6 disponíveis: $3,4 \times 10^{38}$
- Mesmo assim, em um primeiro momento, a cadeia de AS e infraestrutura de roteamento estão preparadas para lidar com a implementação do novo protocolo

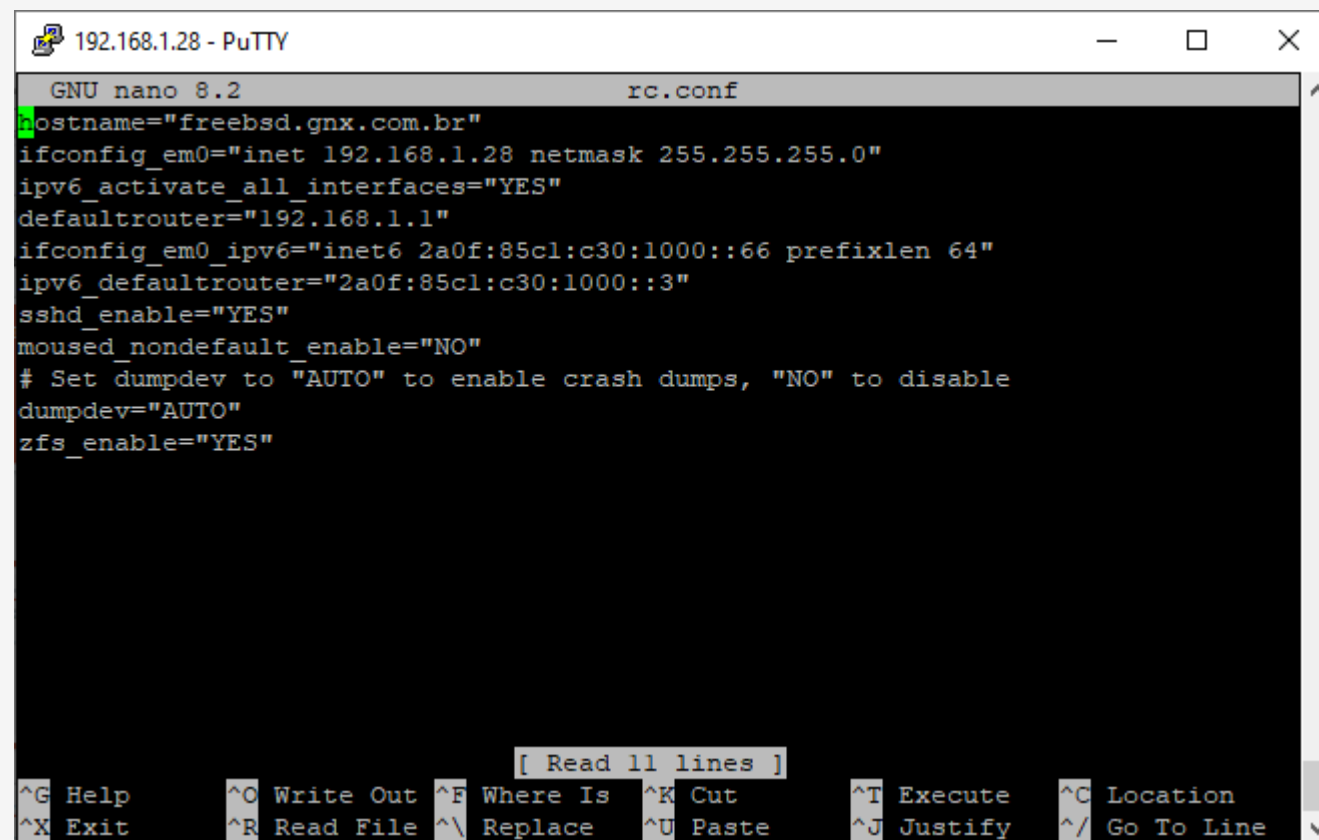


Caso de Uso do FreeBSD como provedor de serviços em IPv6

A grande importância do FreeBSD, é ajudar a fomentar o IPv6, não apenas provendo a conectividade aos usuários, e sim, provendo serviços.

Neste caso, vamos prover um autêntico servidor FreeBSD com IPv6, e os serviços de web e mail, realidade extremamente comum do sistema.

Vamos neste cenário, ter o Apache e o Postfix Instalado, para que os serviços sejam expostos para demonstração



```
192.168.1.28 - PuTTY
GNU nano 8.2 rc.conf
hostname="freebsd.gnx.com.br"
ifconfig_em0="inet 192.168.1.28 netmask 255.255.255.0"
ipv6_activate_all_interfaces="YES"
defaultrouter="192.168.1.1"
ifconfig_em0_ipv6="inet6 2a0f:85c1:c30:1000::66 prefixlen 64"
ipv6_defaultrouter="2a0f:85c1:c30:1000::3"
sshd_enable="YES"
moused_nondefault_enable="NO"
# Set dumpdev to "AUTO" to enable crash dumps, "NO" to disable
dumpdev="AUTO"
zfs_enable="YES"

[ Read 11 lines ]
^G Help      ^O Write Out  ^F Where Is   ^K Cut        ^T Execute    ^C Location
^X Exit      ^R Read File  ^\ Replace    ^U Paste       ^J Justify    ^_ Go To Line
```

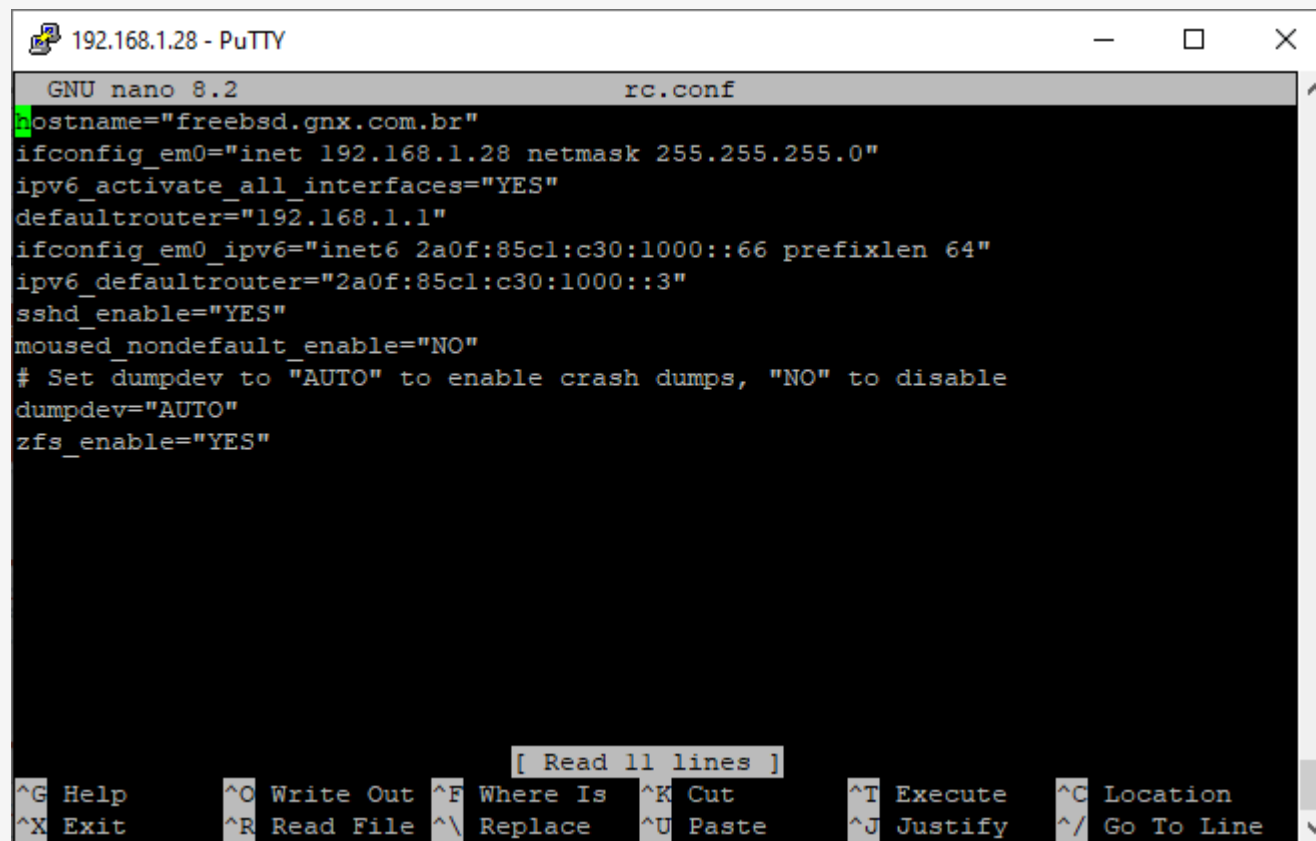


Testando na prática – Apache – Porta 80

Basicamente aqui, fizemos questão de desativar o hostname em IPv4, obrigando qualquer dispositivo a acessar o host em IPv6

Observe o telnet na porta 80. Simples, não?

Rápido, sem complicação, apenas sendo necessária uma adequação das regras de firewall da própria máquina ou do firewall UTM (like pfSense ou OpnSense), para tratar de forma segura, as solicitações



```
192.168.1.28 - PuTTY
GNU nano 8.2 rc.conf
hostname="freebsd.gnx.com.br"
ifconfig_em0="inet 192.168.1.28 netmask 255.255.255.0"
ipv6_activate_all_interfaces="YES"
defaultrouter="192.168.1.1"
ifconfig_em0_ipv6="inet6 2a0f:85c1:c30:1000::66 prefixlen 64"
ipv6_defaultrouter="2a0f:85c1:c30:1000::3"
sshd_enable="YES"
moused_nondefault_enable="NO"
# Set dumpdev to "AUTO" to enable crash dumps, "NO" to disable
dumpdev="AUTO"
zfs_enable="YES"

[ Read 11 lines ]
^G Help      ^O Write Out ^F Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line
```

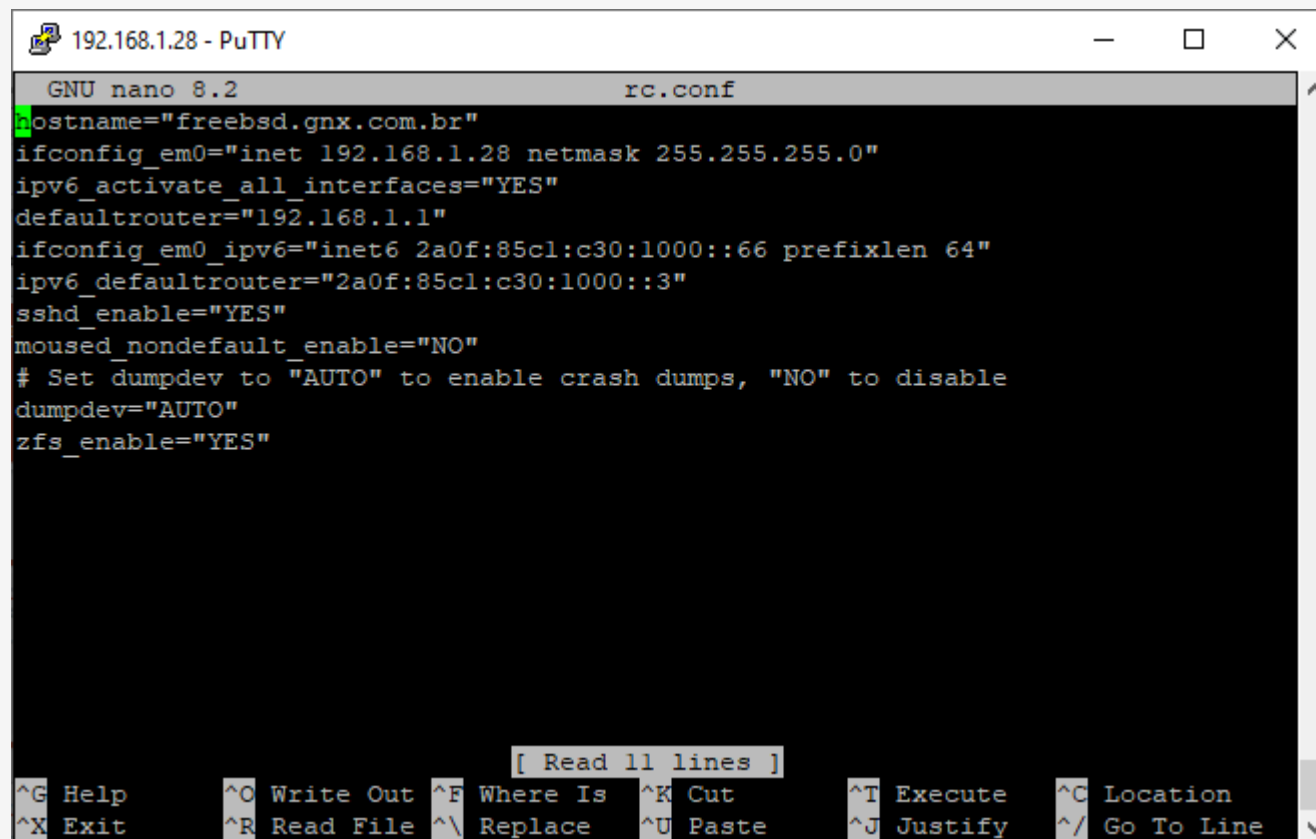


Testando na prática – Postfix – Porta 25

Apenas com o postfix instalado por padrão, vamos testar o uso do mesmo, tendo como resultado a partir da abertura da porta 25, o banner do postfix do servidor.

Curiosidade: menos de 43% dos servidores de e-mail, estão com IPv6 ativado, com uso em produção.

Para serviços de e-mail, quase todas as regras do IPv4, continuam valendo, sendo recomendado o host possuir um DNS Reverso, como é o caso deste FreeBSD



The screenshot shows a PuTTY terminal window titled "192.168.1.28 - PuTTY". Inside, the GNU nano 8.2 editor is open to the file rc.conf. The configuration includes settings for hostname, network interfaces (em0) for both IPv4 and IPv6, default routers, SSH, mouse, crash dumps, and ZFS. The bottom status bar shows "[Read 11 lines]" and a list of keyboard shortcuts for nano editor operations.

```
GNU nano 8.2 rc.conf
hostname="freebsd.gnx.com.br"
ifconfig_em0="inet 192.168.1.28 netmask 255.255.255.0"
ipv6_activate_all_interfaces="YES"
defaultrouter="192.168.1.1"
ifconfig_em0_ipv6="inet6 2a0f:85c1:c30:1000::66 prefixlen 64"
ipv6_defaultrouter="2a0f:85c1:c30:1000::3"
sshd_enable="YES"
moused_nondefault_enable="NO"
# Set dumpdev to "AUTO" to enable crash dumps, "NO" to disable
dumpdev="AUTO"
zfs_enable="YES"
```

[Read 11 lines]

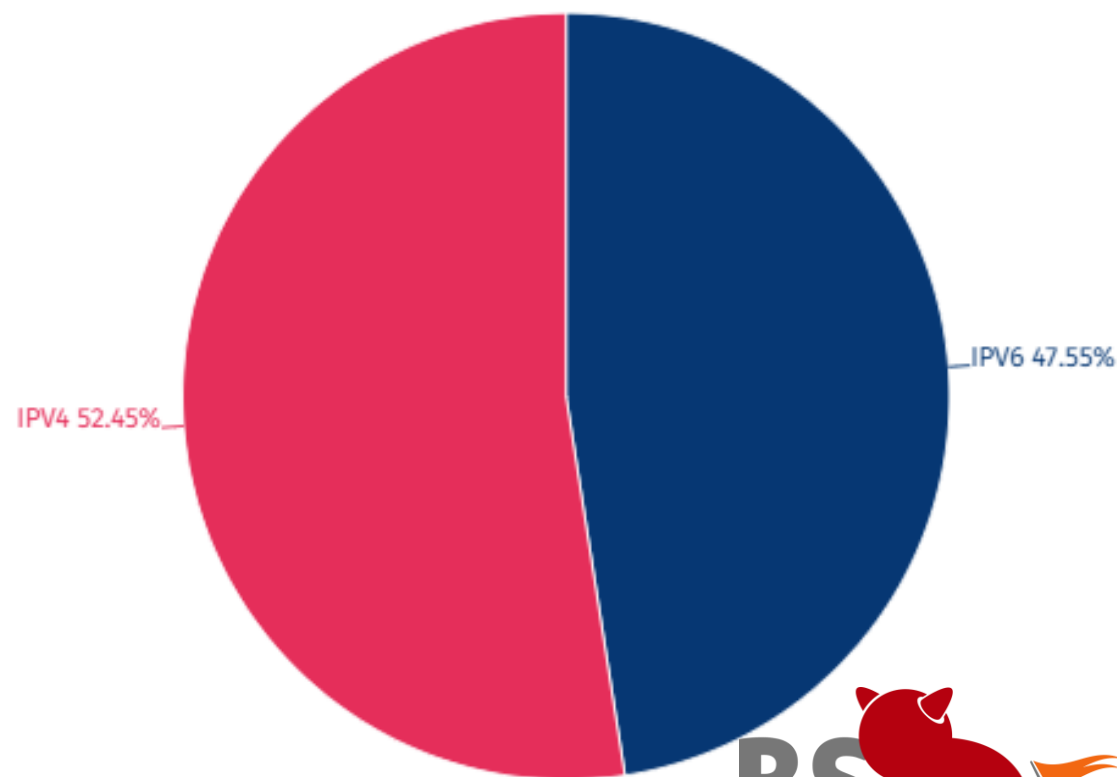
^G Help	^O Write Out	^F Where Is	^K Cut	^T Execute	^C Location
^X Exit	^R Read File	^_ Replace	^U Paste	^J Justify	^_ Go To Line



Motivos para impulsionar o FreeBSD em IPv6

Enquanto 52.45% dos usuários usam IPv4, 47.55% usam IPv6, possivelmente em pilha dual-stack

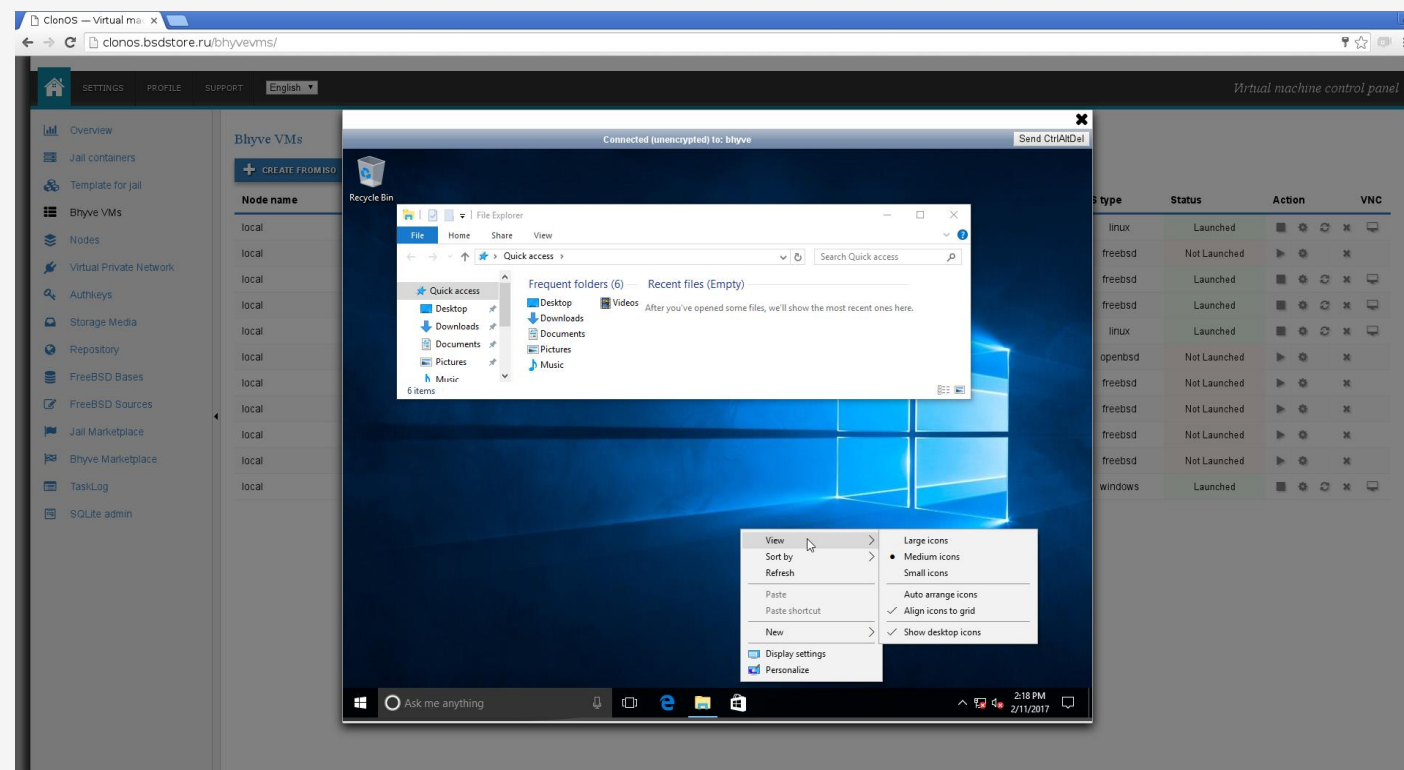
- 1 MAS, as bigtechs começaram a precificar de forma preocupante, os Ips públicos em IPv4, para forçar a migração para IPv6.
- 2 A grande preocupação de vários players de serviços de mercado, é a de que muito em breve, determinados serviços serão ofertados APENAS em IPv6
- 3 O que de CERTA forma, traz luz ao FreeBSD, com as várias melhorias percebidas em relação a conectividade IPv6, em seu último release



IPv6 lembra virtualização que...

Traz o projeto CBSD/Clonos, que está operacional, permitindo criar jails e máquinas virtuais, sendo que estas podem em ambos os protocolos, com livre escolha do administrador.

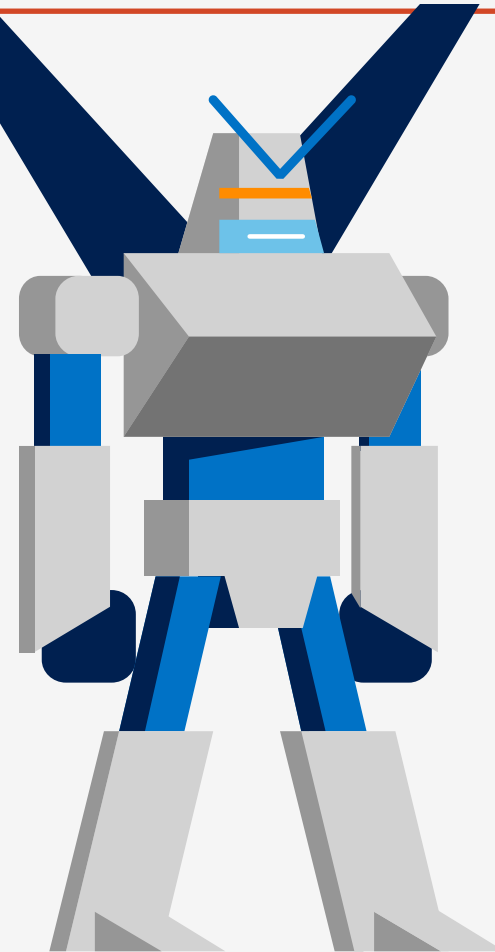
Enquanto o CBSD é o framework que permite criar e administrar jails e VMS no FreeBSD, operando com bhyve, Qemu, Xen e jails, o Clonos, é a interface web para o CBSD



Vamos ver tudo em ação?

Veja agora, o funcionamento do Clonos, com uma jail em IPv6, usando o FreeBSD como Host, demonstrando definitivamente que o FreeBSD pode de várias formas, prover um ambiente pleno com as últimas tendências da tecnologia.

BORA FREEBSD!



Mas ainda falta muita coisa...

Ainda que no dia 6 de junho de 2012, tenhamos a data oficial do lançamento do IPv6 de forma maciça na grande rede, e a RFC 8200 ratifique o IPv6 como um padrão da Internet, a migração efetiva ainda pode levar mais de vinte anos.

Porém, com o pioneirismo do FreeBSD, é possível garantir que as demandas de disponibilidade, serão atendidas por parte do sistema, trazendo ainda mais segurança para o usuário, diante das notáveis melhorias que o IPv6 traz para a conectividade do usuário.



Na parte de conectividade de usuários

Projetos baseados em FreeBSD, como pfSense® e OpnSense®, garantem a conectividade dos usuários, trazendo o Ipv6 para a última milha, ou estruturas de pequeno/médio porte

Mesmo que tenham diferenças estruturais em seus projetos, a base instalada de sistemas, é consideravelmente grande.

Além do que, os serviços em SLAAC e DHCPv6, funcionam sem problemas, entregando a conectividade esperada



Qual é a nossa responsabilidade?

Divulgar intensamente as vantagens do FreeBSD, e principalmente a garantia de performance do IPv6 em diferentes cenários, conquistando o espaço no mercado por si mesmo.

Fomentar, dentro das estruturas atuantes, o uso do IPv6, seja em dual-stack ou nativamente, pois na França por exemplo, medidores de energia que usam G3-PLC, desde 2011 , garante a medição do consumo de mais de 35 milhões de unidades.



É difícil mudar de vez o cenário?

A principal desculpa dos envolvidos no processo de modernização da Internet, até 2012, era de fato a falta de suporte dos equipamentos para o IPv6, o que de fato, é barreira superada.

Ainda que os ISPs disponibilizem redes /64 para usuários finais, grande parte dos técnicos e usuários, não sabem trabalhar com IPv6, que apesar de compartilhar alguma lógica com IPv4, não é **IGUAL**.



Educar, é preciso

As universidades, ainda focam pouco dentro dos laboratórios, em capacitar quanto ao uso do IPv6, sendo que parte significativa das estruturas das próprias universidades, não estão com o IPv6 ativo.

Não é a toa que, pequenas estruturas, que são parte significativa da internet e que fornecem diversos serviços, estão somente em IPv6.
É unicamente por desconhecimento e falta de treinamento



Educar, é preciso (e há solução)

Um curso realmente recomendado , é o do NIC.Br, que tem uma didática leve e realmente aprofundada em relação ao tema do IPv6, trazendo o conhecimento necessário para implementar o protocolo.

Inclusive, o novo curso EAD promete muitas novidades, e terá importante contribuição no segmento de redes



E, os sistemas legados?

Para os sistemas legados, a melhor opção continua a ser, dentre várias, o uso de proxies, como o nginx, o caddy, entre outros, recebendo a conexão em IPv6 e repassando para o dispositivo que somente funciona em IPv4, isso, se for um dispositivo que somente vá receber a conexão.

E, como citamos em algum momento desta palestra, o NAT64, garantindo a nova abordagem.



E por incrível que pareça....

Quando estudamos a implementação do IPv6 e dos serviços que foram demonstrados, lembrando que a maioria deles possui implementação em IPv6, ela é MAIS FÁCIL que em algumas distribuições Linux. É uma constatação imparcial.

Só não é mais fácil que a do Windows, que fornece uma interface GUI para o usuário usar da melhor forma o protocolo



Referências Bibliográficas

Handbook FreeBSD – <https://freebsd.org>

ClonOS – <https://clonos.convectix.com>

BSDSul – <https://bsdsul.com.br>

RFC 2460 - <https://datatracker.ietf.org/doc/html/rfc2460>

NIC.Br – <https://nic.br>



Muito obrigado !

Mauro Paes Corrêa

Fundador do BSDSul

<https://bsdsul.com.br>

CEO – GNX

<https://gnx.com.br>

Linkedin – <https://linkedin.com/in/mauopaescorrea>

E-Mail: mauro@gnx.com.br

