Implementação de servidores recursivos (BIND e Unbound) com DNSSEC e Hyperlocal

Semana de Capacitação CEPTRO.br – Edição Online

Guia de práticas
26 Agosto 2020

ICANN

TABLE OF CONTENTS

Considerações iniciais

Muito obrigado por participar deste curso. Seja bem-vindo!
Este guia foi desenvolvido como material de apoio didático para a Semana de Capacitação Edição Online promovida pelo NIC.br. O objetivo deste conteúdo é demonstrar de maneira prática e simples a aplicação dos conceitos de DNSSEC e Hyperlocal em servidores de DNS recursivos. Este material deve ser utilizado exclusivamente para uso educacional. As configurações aqui apresentadas **não são recomendações para uso em ambiente de produção** em função da simplificação didática utilizada. Em caso de dúvidas e maiores informações, por favor entre em contato com daniel.fink@icann.org e nicolas.antoniello@icann.org .
Bom treinamento !

Preparando o seu computador

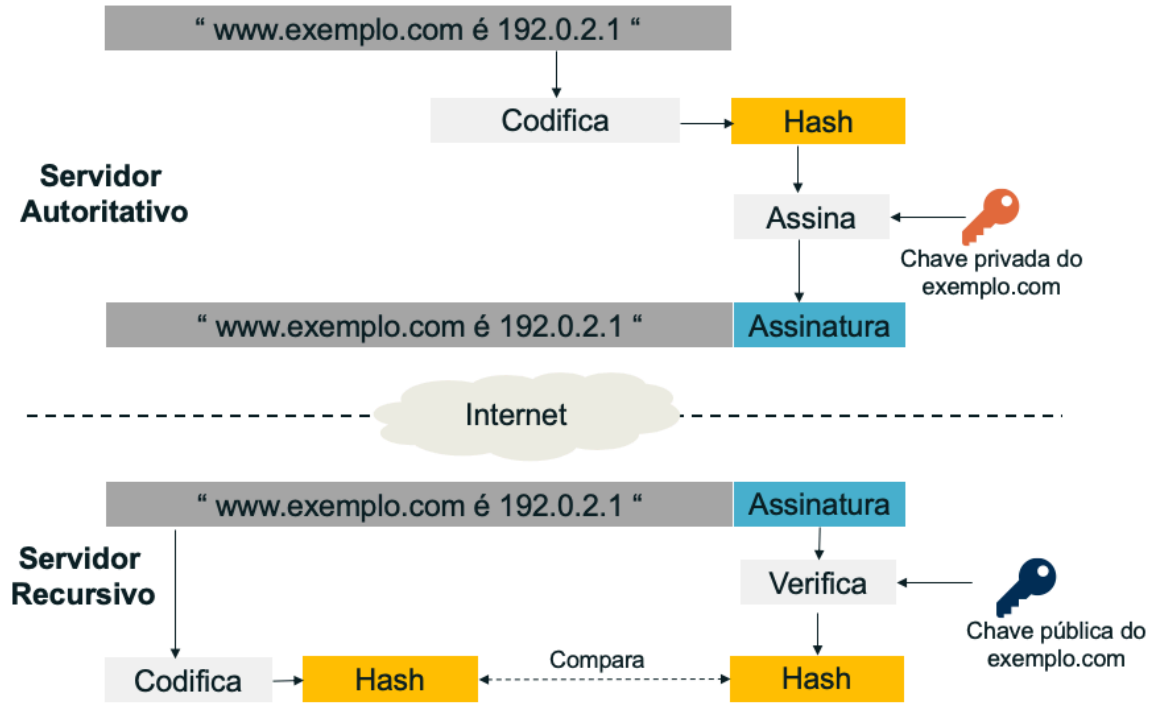
Se desejar saber como preparar o seu computador para este treinamento, acesse: https://drive.google.com/file/d/14zocTDt2cLQYyeSwhkYtdJ8xgmLjjHdd/view?usp=sharing

Parte 1: Conceitos que serão aplicados nesta prática

Aqui temos um breve resumo dos dois conceitos que iremos implementar nesta prática: DNSSEC e Hyperlocal. Confira a seguir.

DNSSEC: Extensões de Segurança do DNS

Quando o DNS foi criado no início dos anos 80, a segurança não era a principal prioridade. Era possível que um invasor, em raros casos, direcionar a consulta de um usuário para outro destino. A solução para essa vulnerabilidade foi apresentada nos anos 90, quando a comunidade técnica do DNS lançou o DNSSEC (DNS Security Extensions) ou Extensões de Segurança do DNS. A RFC3833 conta este histórico e as RFC4033 , RFC4034 e RFC4035 trazem os detalhes técnicos. Em resumo, o DNSSEC funciona assim:



Nesta ilustração um determinado servidor recursivo solicita o número IP do site www.example.com para o servidor autoritativo. Antes de enviar a resposta, que seria "www.example.com é 192.0.2.1", o servidor autoritativo transforma o conteúdo da resposta em um código numérico de tamanho fixo (chamado Hash) e criptografa este código com uma chave privada gerando uma assinatura. Esta assinatura será enviada juntamente com a resposta pela internet. O servidor recursivo recebe a resposta "www.example.com é 192.0.2.1" e a assinatura. Então o recursivo faz duas operações distintas: 1) gera a sua própria "Hash" da resposta através da mesma codificação utilizada pelo autoritativo e; 2) Utiliza uma chave pública para gerar uma segunda "Hash" com base na assinatura recebida da transmissão. Se as duas "Hash's" forem iguais, a informação é autêntica.

Para saber mais: https://wiki.brasilpeeringforum.org/w/DNSSEC_Seguranca_do_DNS

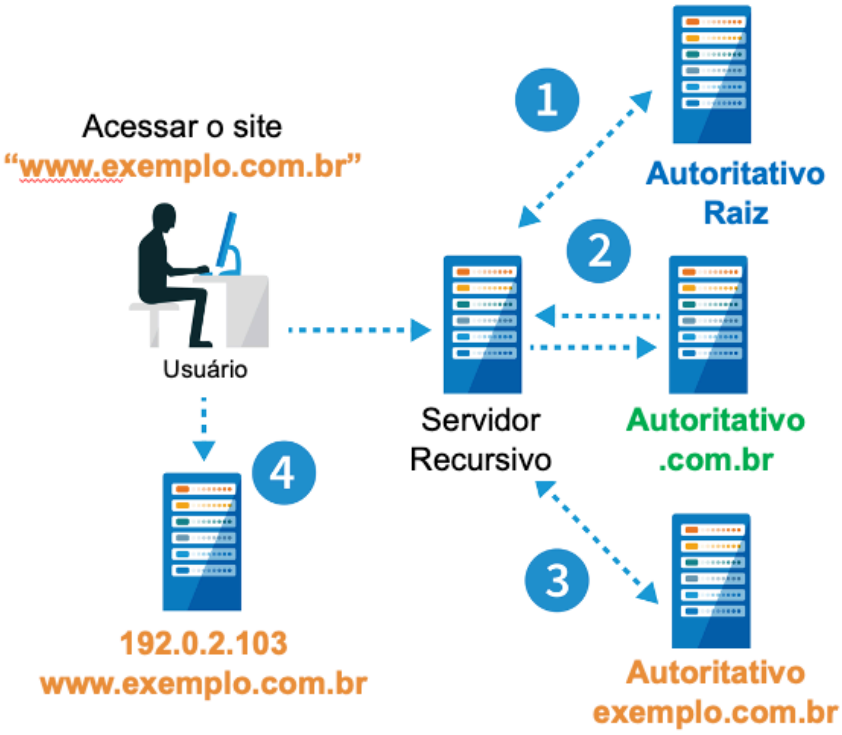
Hyperlocal: Servidor raiz local no recursivo

A Zona Raiz do Sistema de Nomes de Domínios (DNS) é servida por 12 organizações que operam instâncias anycast de servidores de nomes autoritativos provendo respostas para a raiz do DNS. Estas instâncias estão distribuídas em mais de 1000 localidades ao redor do mundo. Apesar deste grande número de servidores e alta capacidade provisionada para a resolução da raiz de nomes, ainda existe a possibilidade de que um grande ataque coordenado de negação de serviço (DDoS) possa comprometer o acesso à internet para muitos usuários.

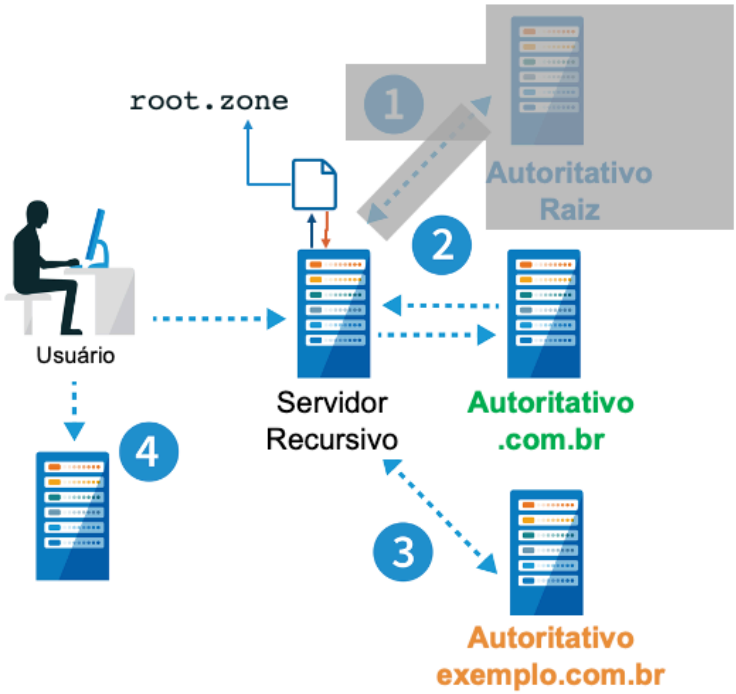
Para minimizar e prevenir esta ameaça, existe a alternativa de adicionar um fator de resiliência na configuração dos servidores recursivos do provedor de internet através do uso de uma cópia local da zona raiz, chamada de Hyperlocal.

Hyperlocal é apresentado em detalhes na RFC8806 e, resumidamente, **consiste em executar uma cópia da zona raiz no mesmo servidor de serviços de resolução recursiva.** Desta forma, as consultas à zona raiz dos clientes são respondidas localmente sem necessidade comunicação externa entre os servidores. Isso resulta em maior robustez do serviço em caso de ataques e ganhos na velocidade de provimento de respostas às consultas ao DNS dos usuários.

Para saber mais: https://wiki.brasilpeeringforum.org/w/Tutorial_DNS_Hyperlocal e https://tools.ietf.org/html/rfc8806



Resolução de DNS sem Hyperlocal



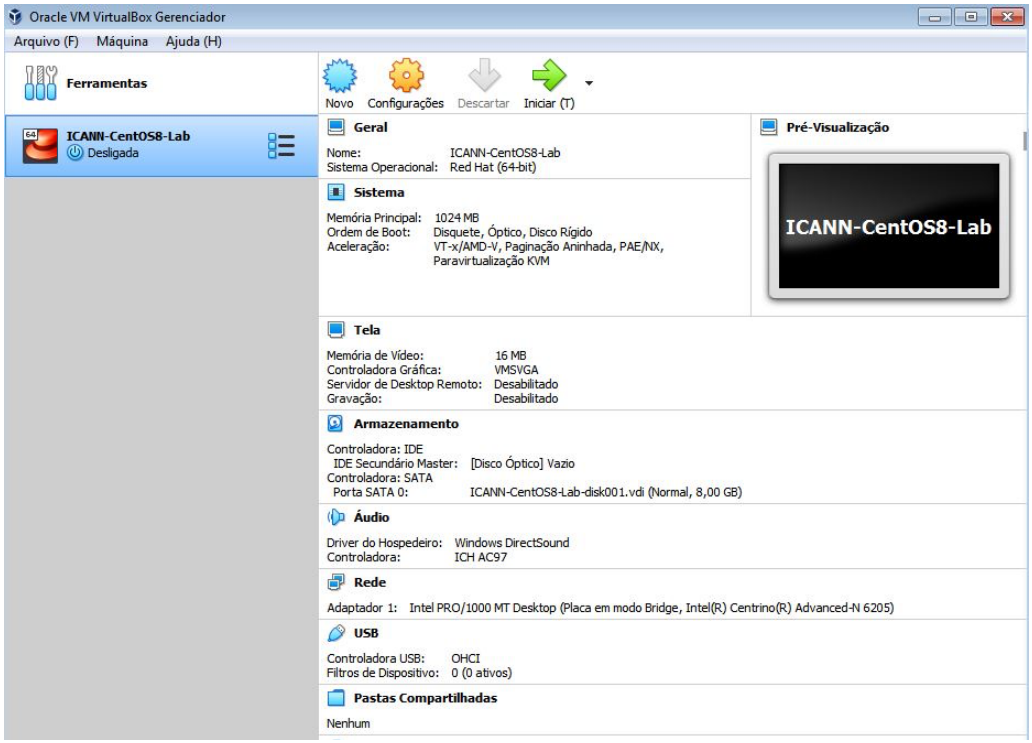
Resolução de DNS com Hyperlocal

Parte 2: CentOS8 com Unbound

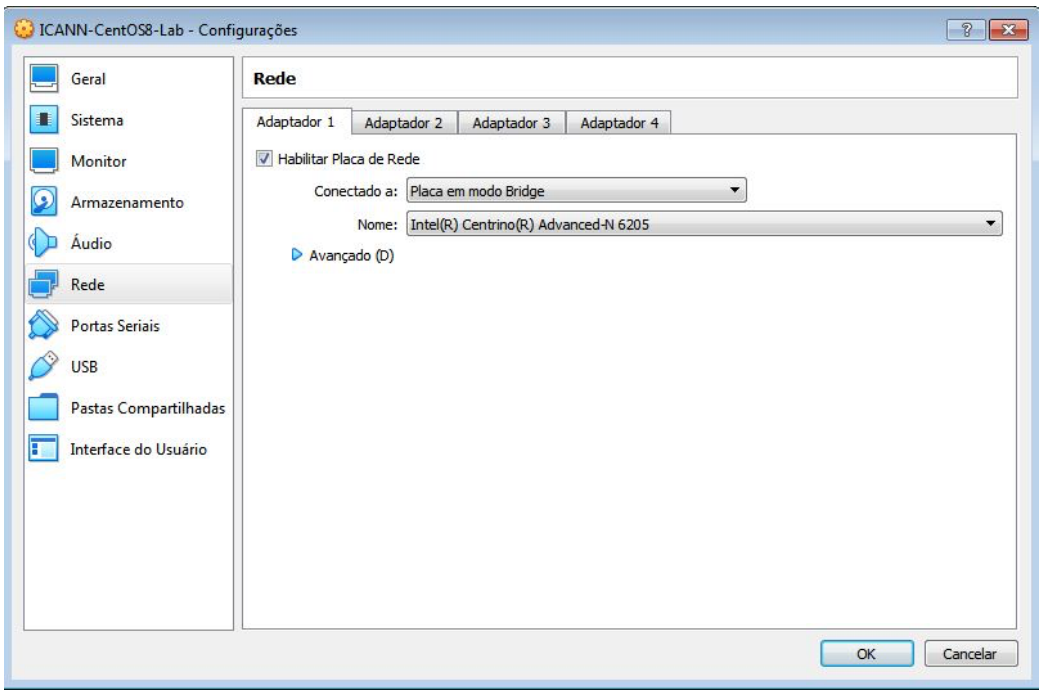
Esta é a primeira parte da prática onde iremos configurar um servidor recursivo Unbound no sistema operacional CentOS8.
Para mais informações sobre o Unbound, visite: https://nlnetlabs.nl/unbound

Iniciando o CentOS no VirtualBox

- Selecione a máquina virtual **ICANN-CentOS8-Lab** e clique em Configurações.

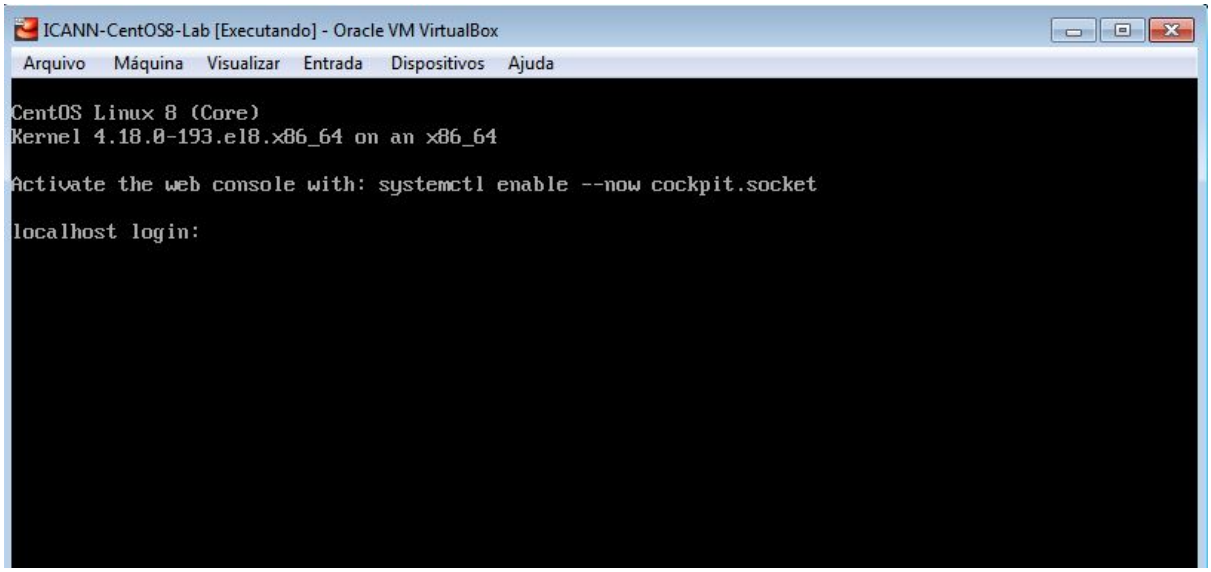


- Clique na tab “Rede” e selecione “Conectado a: **Placa em modo Bridge**”. Clique em OK.

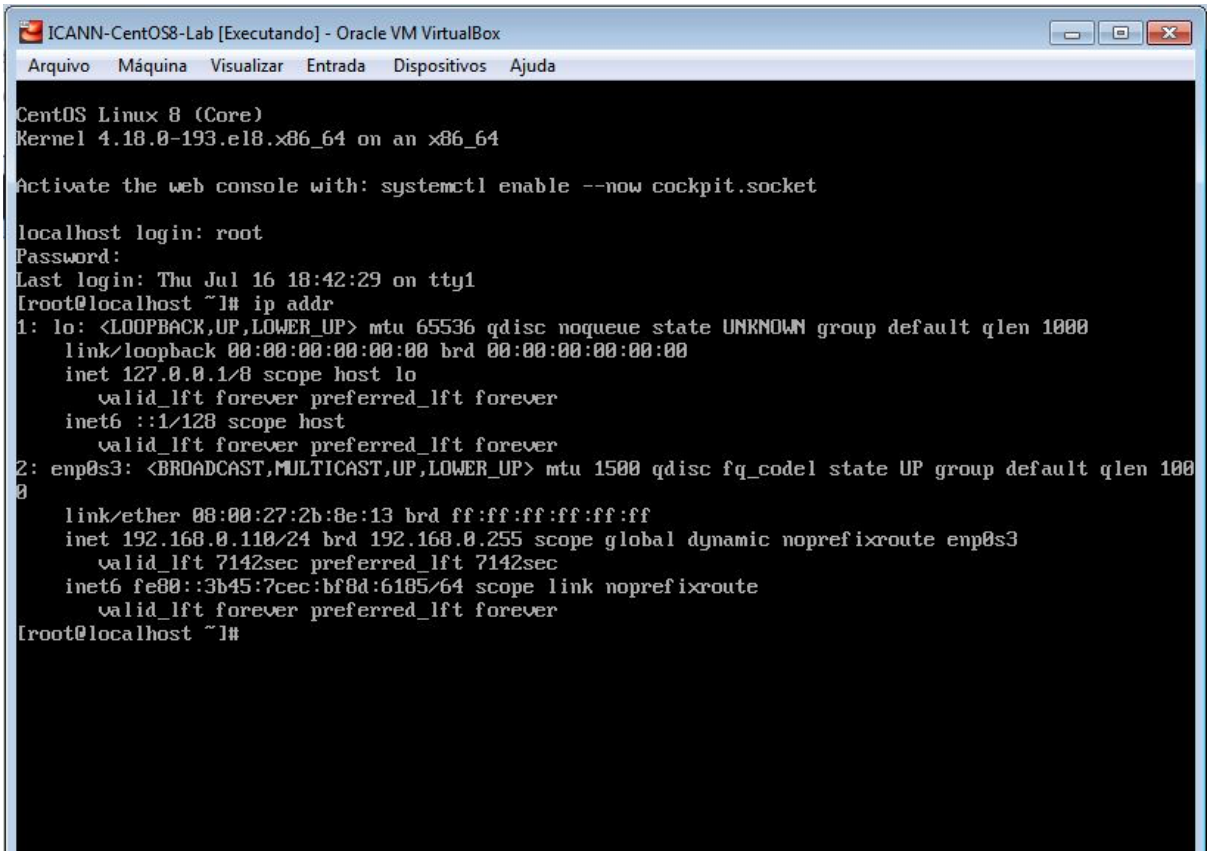


- Clique em Iniciar

- Login: `root`
- Senha: `icannlab2`



- Encontrar o endereço IP da máquina virtual
  `$ ip addr`



Neste caso, o IP é 192.168.0.110. (Rede 192.168.0.0/24)
Anote o seu número IP.

Preparando o Servidor

- Instalação dos pacotes necessários

```
$ yum install nano net-tools wget
$ yum install unbound bind-utils
```

- Desabilitar o FIREWALLD

```
$ systemctl disable firewalld
```

- Desabilitar o SELINUX

```
$ nano /etc/selinux/config
Alterar para SELINUX=disabled

Control X
```

- Reiniciar a máquina

```
$ init 0
```

Preparando o Unbound

- Acessar a pasta do Unbound

```
$ cd /etc/unbound
```

- Remover o arquivo unbound.conf

```
$ rm unbound.conf -f
```

- Remover o arquivo root.key

```
$ rm root.key -f
```

- Baixar o arquivo named.root

```
$ wget https://www.internic.net/domain/named.root
```

- Verifique se foi baixado corretamente

```
$ ls -l
```

PAUSA. Vamos dar uma olhada neste arquivo.

```
$ nano named.root

; Esse arquivo contém as informações dos servidores de nomes raiz
necessárias para inicializar o cache dos servidores de nomes de
domínio da Internet.
; FORMERLY NS.INTERNIC.NET
;
.                        3600000 NS    A.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET. 3600000 A     198.41.0.4
A.ROOT-SERVERS.NET. 3600000 AAAA  2001:503:ba3e::2:30.
        ...
; OPERATED BY ICANN
;
.                       3600000 NS    L.ROOT-SERVERS.NET.
L.ROOT-SERVERS.NET. 3600000 A     199.7.83.42
L.ROOT-SERVERS.NET. 3600000 AAAA  2001:500:9f::42
;
; OPERATED BY WIDE
;
.                       3600000 NS    M.ROOT-SERVERS.NET.
M.ROOT-SERVERS.NET. 3600000 A     202.12.27.33
```

```
M.ROOT-SERVERS.NET. 3600000 AAAA  2001:dc3::35
```

- Editando o arquivo de configuração do Unbound

i. Criar um novo arquivo unbound.conf (lembre-se que apagamos o original). Escrever a configuração abaixo.

```
$ nano unbound.conf
```

```
server:
      directory: "/etc/unbound"
      username: unbound

      interface: 0.0.0.0

      access-control: 0.0.0.0/0 refuse
      access-control: 192.168.0.0/24 allow
      access-control: 127.0.0.0/8 allow

      port: 53

      do-udp: yes
      do-tcp: yes
      do-ip4: yes
      do-ip6: no
```

 Ajuste a faixa de IPs da sua rede

- Salve o arquivo: Control X

- Verifique se existem erros no arquivo

```
$ unbound-checkconf
```

- Vincular os arquivos para o usuário unbound

```
$ ls -l
```

→ Veja que alguns arquivos estão vinculados ao usuário root e outros ao usuário unbound.

```
$ chown unbound.unbound /etc/unbound/ -R
```

```
$ ls -l
```

→ Verifique se todos arquivos estão vinculados ao usuário unbound (vamos usar este comando algumas vezes mais para frente)

- Habilitar o Unbound

```
$ systemctl enable unbound
```

- Iniciar o Unbound

```
$ systemctl start unbound
```

- Verificar o funcionamento

```
$ systemctl status unbound
```

- Primeiro teste

```
$ dig @<IP do seu servidor> nic.br
```

Parabéns ! Unbound funcionando !

- Ajustando o resolv.conf para facilitar os exercícios.
    - Abrir o /etc/resolv.conf e incluir o endereço de loopback

```
$ nano /etc/resolv.conf

nameserver 127.0.0.1
```

Control X

O comando DIG

Pequeno tutorial sobre o DIG:

Comando DIG

- ***Domain Information Groper***
  - Útil para troubleshooting e treinamentos
- Componente do pacote BIND (instalamos o bind-utils, lembra?)

```
$ dig [servidor] [nome] [tipo]
```

- Digite apenas dig no terminal

```
$ dig
```

- O que apareceu?
- Qual foi o tempo de resposta (Query time) ?

- Faça alguns testes com o dig

```
$ dig o-seu-site-favorito a
$ dig www.google.com a
$ dig lacnic.net mx
$ dig dominioquenaoexiste.algumtld any
$ dig afrinic.net aaaa
```

- Agora envie consultas pra outro resolver.

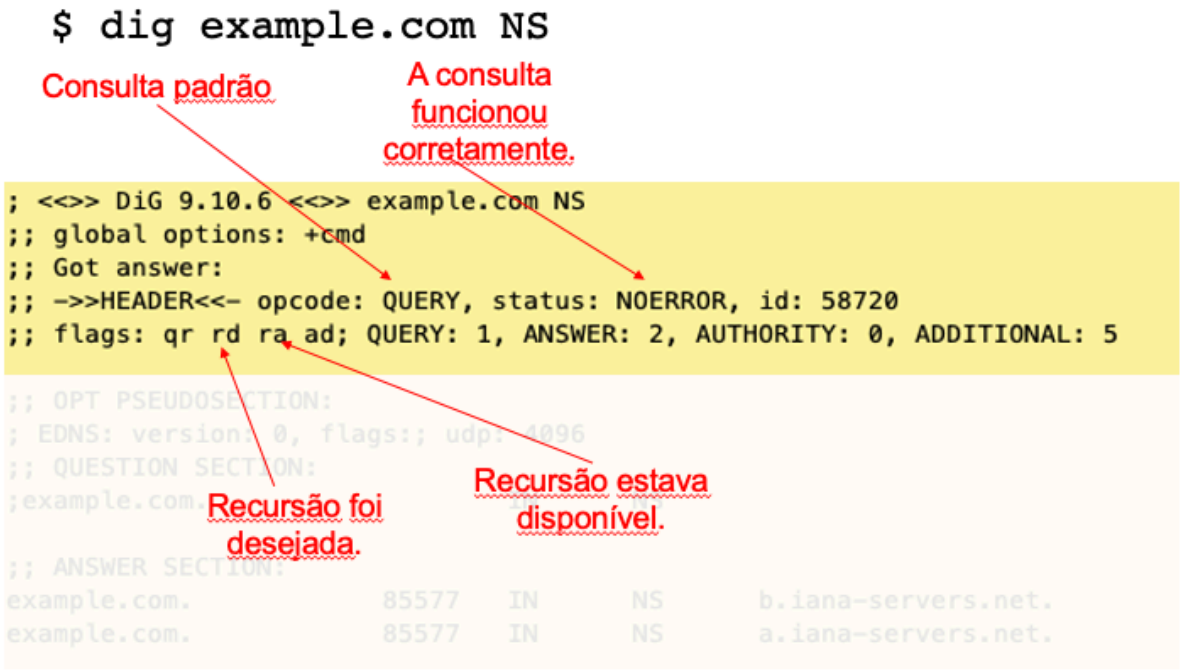
```
$ dig @8.8.8.8 o-seu-site-favorito a
$ dig @9.9.9.9 www.google.com a
```

Outros testes. Compare os tempos de resposta.

- Verificar o tempo de resposta dos servidores raiz

```
$ dig @l.root-servers.net id.server ch txt

$ dig @b.root-servers.net id.server ch txt
```

- Agora vamos decifrar um pouco mais a resposta do DIG.



- Teste de DNSSEC

```
$ dig dnssec-failed.org
```

O que acontece?

Implementando o DNSSEC no Unbound

- Garantir que o arquivo root.key foi removido do /etc/unbound (já fizemos isso acima)

```
$ rm root.key -f
```

- Gerar um arquivo root.key atualizado

```
$ unbound-anchor -a /etc/unbound/root.key -v

$ chown unbound.unbound /etc/unbound/ -R

$ ls -l
```

- Abrir o unbound.conf

```
$ nano unbound.conf
```

Adicionar a partir do #DNSSEC:

```
server:
```

```
directory: "/etc/unbound"
username: unbound

interface: 0.0.0.0

access-control: 0.0.0.0/0 refuse
access-control: 192.168.0.0/24 allow
access-control: 127.0.0.0/8 allow

port: 53

do-udp: yes
do-tcp: yes
do-ip4: yes
do-ip6: no

#DNSSEC
auto-trust-anchor-file: "/etc/unbound/root.key"
```

Control X

```
$ unbound-checkconf

$ systemctl restart unbound
```

- Testando o DNSSEC

```
$ dig nic.br +dnssec +multi
```
(observe a flag ad: dado autenticado)

```
$ dig dnssec-failed.org
```
(Qual é o status?)

- Verificando os Resource Records do DNSSEC

DNSKEY (Chave pública do domínio)
```
$ dig nic.br DNSKEY
```

DS (Ponteiro para a cadeia de confiança)
```
$ dig nic.br DS
```

RRSIG (Assinatura do RRset)
```
$ dig nic.br RRSIG
```

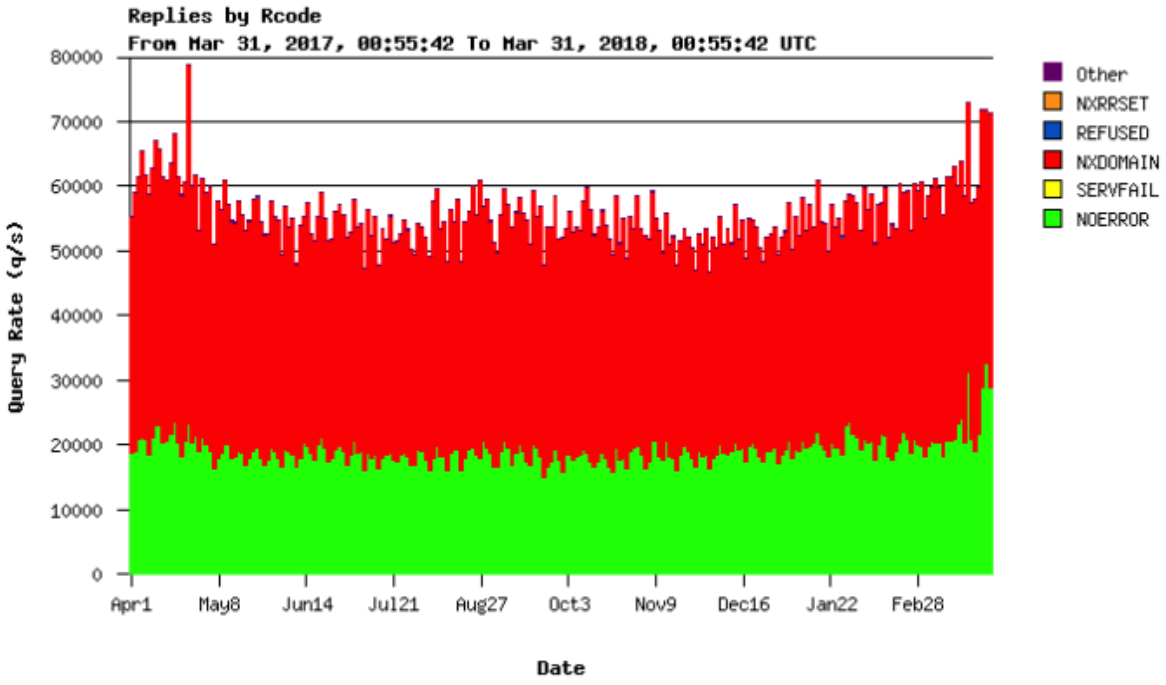
- Como criar uma exceção para o DNSSEC no unbound.conf

```
domain-insecure: "dnssec-failed.org"
```

Implementando o Hyperlocal

2) Curiosidade:

65% da carga dos servidores raiz é dedicada a NXDOMAIN



Yearly graph of return codes on K-Root, provided by the RIPE NCC

Fonte: https://www.ripe.net/analyse/dns/k-root/statistics/?type=ROOT&increment=yearly

- Teste antes da implementação

```
Teste 1: Unbound recursivo sem o Hyperlocal

$ dig domaininvalid.com.br.xxxxxxx
```

- Editar o unbound.conf

```
$ nano unbound.conf
```

```
server:
      directory: "/etc/unbound"
      username: unbound

      interface: 0.0.0.0

      access-control: 0.0.0.0/0 refuse
      access-control: 192.168.0.0/24 allow
      access-control: 127.0.0.0/8 allow

      port: 53

      do-udp: yes
      do-tcp: yes
      do-ip4: yes
      do-ip6: no

      #DNSSEC
      auto-trust-anchor-file: "/etc/unbound/root.key"

      #HYPERLOCAL

      auth-zone:
          name: "."
          master: 192.228.79.201 # b.root-servers.net
          master: 192.33.4.12      # c.root-servers.net
          master: 192.5.5.241      # f.root-servers.net
          master: 192.112.36.4     # g.root-servers.net
          master: 193.0.14.129     # k.root-servers.net
          master: 192.0.47.132     # xfr.cjr.dns.icann.org
          master: 192.0.32.132     # xfr.lax.dns.icann.org
          master: 2001:500:84::b   # b.root-servers.net
          master: 2001:500:2f::f   # f.root-servers.net
          master: 2001:7fd::1      # k.root-servers.net
          master: 2620:0:2830:202::132 # xfr.cjr.dns.icann.org
          master: 2620:0:2d0:202::132  # xfr.lax.dns.icann.org
          fallback-enabled: yes
          for-downstream: no
          for-upstream: yes
          zonefile: "root.zone"
```

Control X

```
$ unbound-checkconf

$ systemctl restart unbound
```

Aguarde alguns minutos para carregar o root.zone.
Verifique se o arquivo root.zone foi criado com o:

```
$ ls -l
```

Verifique o conteúdo do arquivo.

```
$ nano root.zone
```

Testando o Hyperlocal

```
Teste 2: Servidor público Google

$ dig @8.8.8.8 domaininvalid.com.br.xxxxxxx


Teste 3: Recursivo do meu provedor

$ dig @192.168.0.1 domaininvalid.com.br.xxxxxxx


Teste 4: Unbound recursivo com o Hyperlocal

$ dig domaininvalid.com.br.xxxxxxx
```

- Configurando o Firewalld
(Obrigado Canal do Felipe Padilha: https://www.youtube.com/watch?v=GPFvD1sY0xY)

Abrir o arquivo de configuração do Firewalld:

```
$ nano /etc/firewalld/zones/public.xml
```

Acrescentar no final antes do último </zone>

```
<rule family="ipv4">
<source address="192.168.0.0/24"/>
<port protocol="tcp" port="53"/>
<accept/>
</rule>
<rule family="ipv4">
<source address="192.168.0.0/24"/>
<port protocol="udp" port="53"/>
<accept/>
</rule>
```

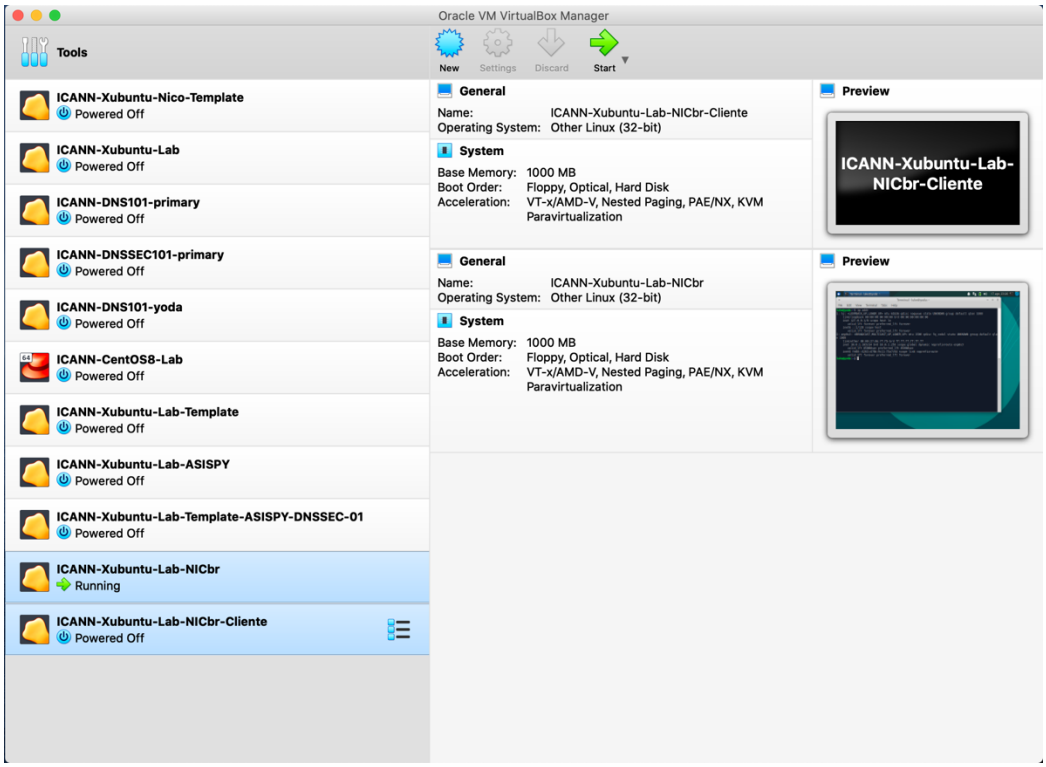
Control X

```
$ systemctl start firewalld

$ firewall-cmd --reload
```

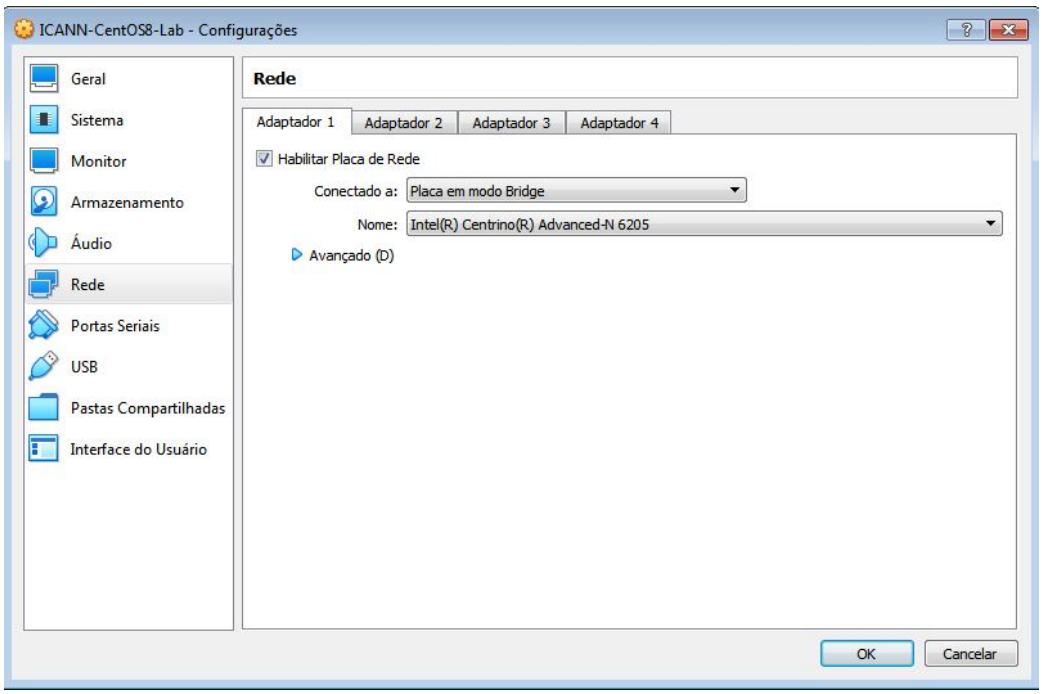
Parte 3: Xubuntu com BIND

Iniciando o Xubuntu no VirtualBox

- Selecione a máquina virtual **ICANN-Xubuntu-Lab** e clique em Configurações.



- Clique na tab “Rede” e selecione “Conectado a: **Placa em modo Bridge**”. Clique em OK.

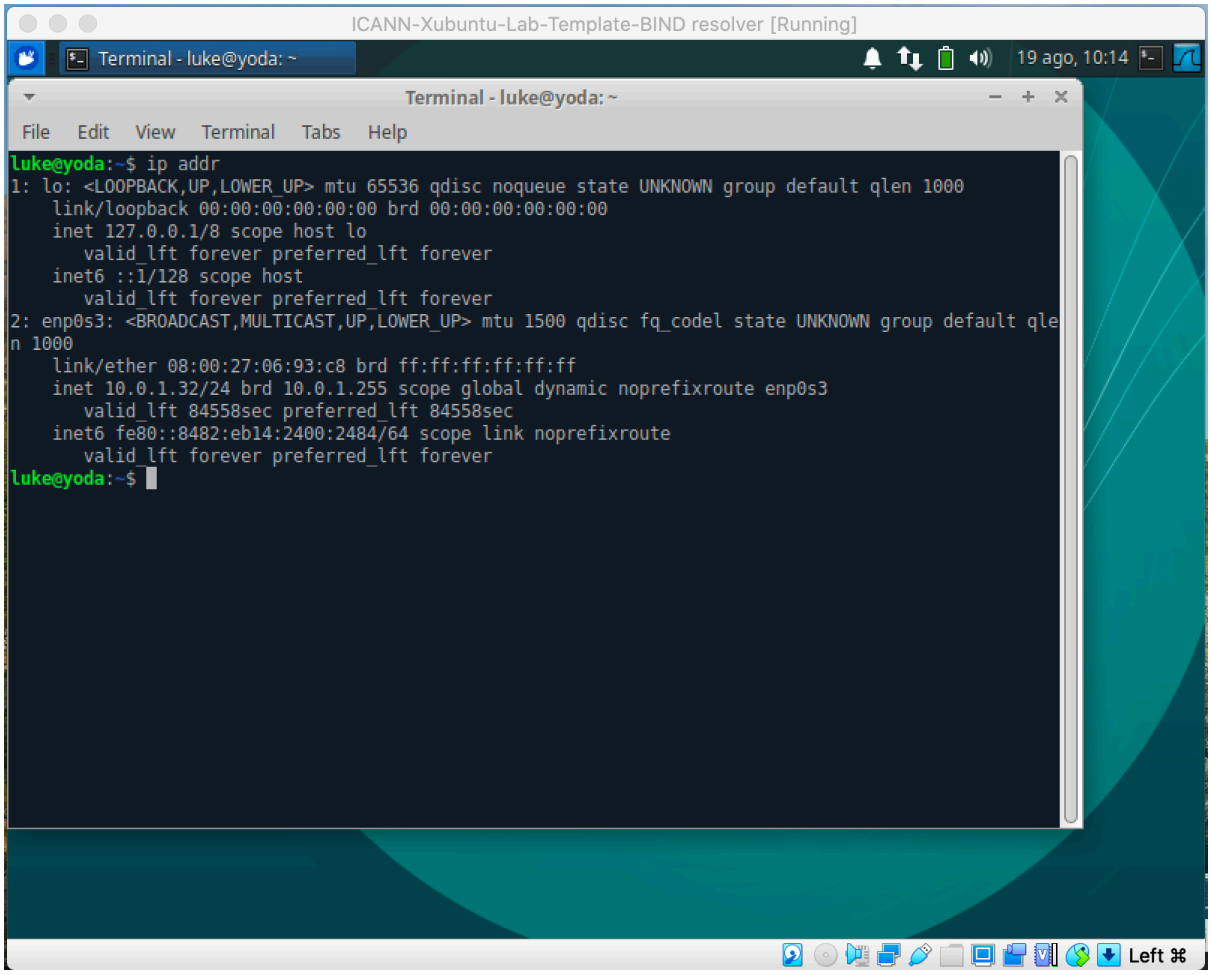


- Clique em Iniciar

- Login: `Luke Skywalker`
- Senha: `icannlab1`

- Encontrar o endereço IP da máquina virtual (chamado “yoda”)

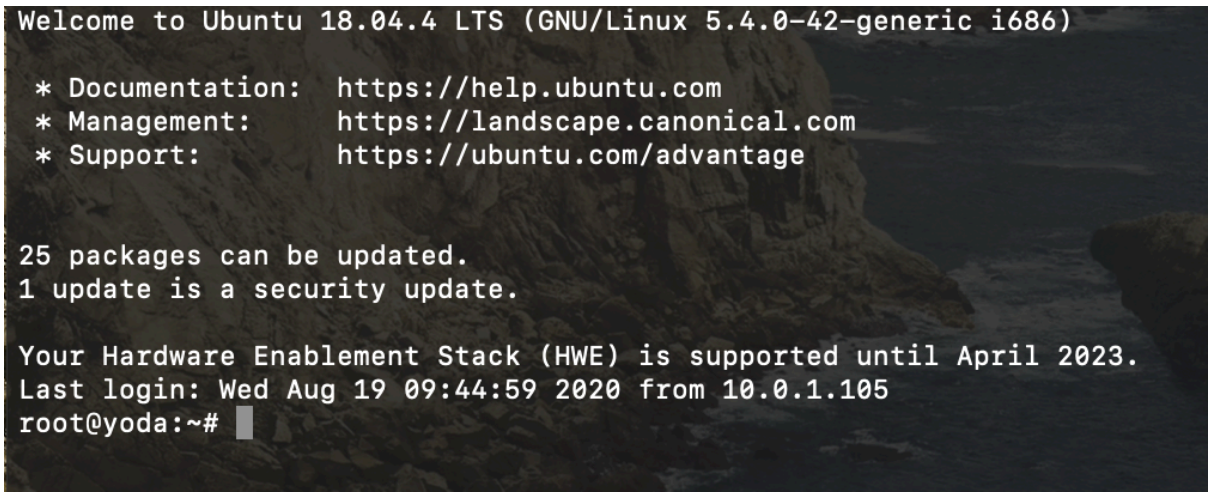
```
# ip addr

10.0.1.103 (Network: 10.0.1.0/24)
```



- Conectando-se ao servidor para instalar e configurar o BIND

Agora, na parte de configuração, usaremos um cliente SSH para se conectar ao servidor (yoda) com privilégios de administrador (root). Para fazer isso, se nossa máquina tiver Linux ou OSX, basta abrir um terminal e conectar ao nosso servidor via SSH:

```
$ ssh root@10.0.1.32
```



Se estivermos usando Windows, podemos usar o aplicativo cliente SSH chamado PUTTY que instalamos anteriormente para nos conectar ao nosso servidor (yoda).

Preparando o servidor

- Instalação de pacotes necessários

```
# apt-get install bind9 bind9utils

# apt-get install bind9-doc dnsutils
```

Para instalar os pacotes necessários para executar o BIND, usaremos os comandos acima (`apt-get`), um de cada vez.
Se o processo de instalação nos perguntar se queremos instalar algum dos pacotes, diremos sim (`yes`).
Os comandos acima instalaram os pacotes BIND, a documentação do BIND, os utilitários BIND (por exemplo, para verificar a configuração, etc) e utilitários DNS.

- É conveniente adicionar uma regra para permitir a porta 53 no caso de o firewall do sistema operacional estar em execução

```
# ufw allow 53
```

Preparando o BIND

- Acessar a pasta do BIND

```
# cd /etc/bind
```

- A instalação padrão trará as informações dos servidores raiz

```
# more db.root
```

```
;       This file holds the information on root name servers needed
;       to initialize cache of Internet domain name servers
;       (e.g. reference this file in the "cache  .  <file>"
;       configuration file of BIND domain name servers).
;
;       This file is made available by InterNIC
;       under anonymous FTP as
;           file                /domain/named.cache
;           on server           FTP.INTERNIC.NET
;       -OR-                    RS.INTERNIC.NET
;
;       last update:    February 17, 2016
;       related version of root zone:   2016021701
;
; formerly NS.INTERNIC.NET
;
.                       3600000      NS    A.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET.     3600000      A     198.41.0.4
A.ROOT-SERVERS.NET.     3600000      AAAA  2001:503:ba3e::2:30
;
; FORMERLY NS1.ISI.EDU
;
…

; OPERATED BY ICANN
;
.                       3600000      NS    L.ROOT-SERVERS.NET.
L.ROOT-SERVERS.NET.     3600000      A     199.7.83.42
L.ROOT-SERVERS.NET.     3600000      AAAA  2001:500:3::42
;
; OPERATED BY WIDE
;
.                       3600000      NS    M.ROOT-SERVERS.NET.
M.ROOT-SERVERS.NET.     3600000      A     202.12.27.33
M.ROOT-SERVERS.NET.     3600000      AAAA  2001:dc3::35
; End of file
```

- Editando o arquivo de configuração do BIND

- O arquivo de configuração (`named.conf.options`) da instalação padrão está localizado no diretório BIND (`/etc/bind`)

```
# more named.conf.options
```

```
options {
     directory "/var/cache/bind";

     // If there is a firewall between you and nameservers you want
     // to talk to, you may need to fix the firewall to allow
     // multiple
     // ports to talk.  See http://www.kb.cert.org/vuls/id/800113

     // If your ISP provided one or more IP addresses for stable
     // nameservers, you probably want to use them as forwarders.
     // Uncomment the following block, and insert the addresses
     // replacing
     // the all-0's placeholder.

     // forwarders {
     //   0.0.0.0;
     // };

     //========================================================
     // If BIND logs error messages about the root key being
     // expired,
     // you will need to update your keys.  See
     // https://www.isc.org/bind-keys
     //========================================================
     dnssec-validation auto;

     auth-nxdomain no;    # conform to RFC1035
     listen-on-v6 { any; };
};
```

- Vamos realizar as seguintes ações na configuração (explicação das opções)
    - Para simplificar a prática, desabilitamos a capacidade de receber consultas usando o protocolo IPv6 (somente consultas usando IPv4). No entanto, recomendamos que qualquer servidor DNS recursivo tenha os protocolos IPv4 e IPv6 configurados em todos os casos de implantações de produção.

      ```
      //listen-on-v6 { any; };
      ```

    - Para esta primeira parte da prática, desabilitaremos o DNSSEC explicitamente. Observe que, por padrão, o BIND terá a validação DNSSEC ativada.

      ```
      dnssec-enable no;
      //dnssec-validation auto;
      ```

      Quando dnssec-validation é definido como automático, o padrão é a zona de raiz do DNS como a âncora de confiança. O BIND inclui uma cópia da chave raiz que é mantida atualizada automaticamente. Se definido como sim, uma âncora de confiança deve ser configurada explicitamente usando a opção de managed-keys ou trusted-keys.

- Ativamos explicitamente a recursão.

  ```
  recursion yes;
  ```

- Criamos uma lista de acesso para permitir apenas consultas DNS do próprio servidor ou da nossa rede (para evitar que o servidor recursivo esteja aberto ao mundo).

  ```
  listen-on port 53 { localhost; 10.0.1.0/24; };
  allow-query { localhost; 10.0.1.0/24; };
  ```

Para que o arquivo de configuração seja o seguinte

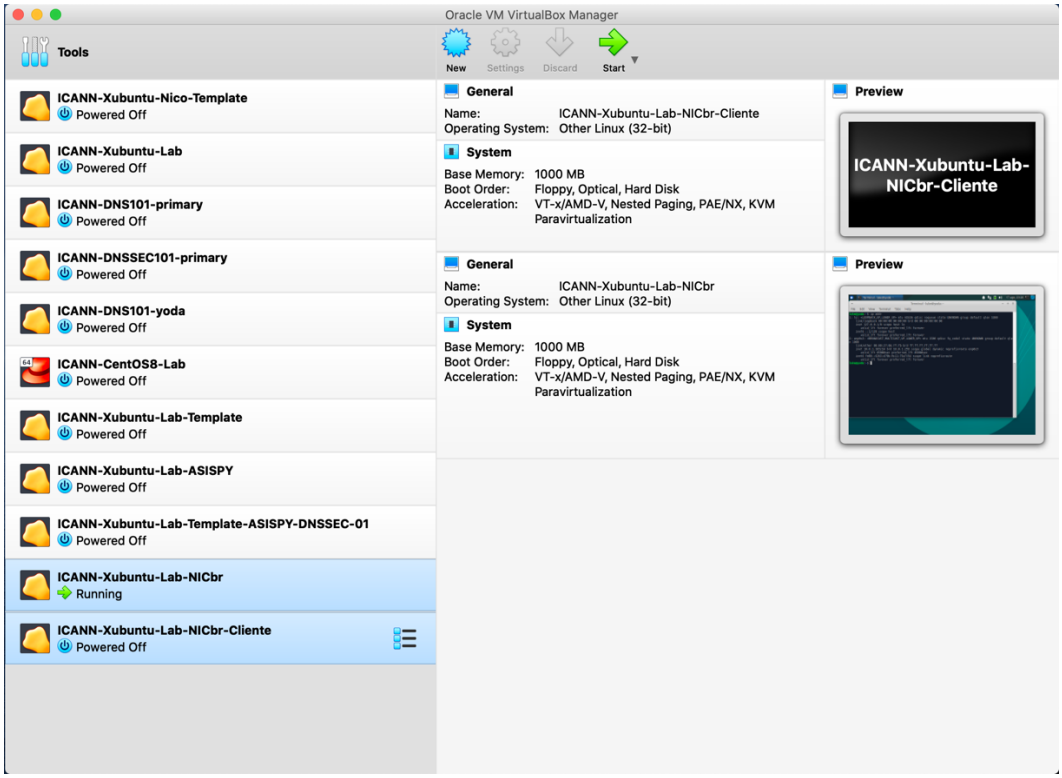
```
# nano named.conf.options

options {
     directory "/var/cache/bind";

     // If there is a firewall between you and nameservers you want
     // to talk to, you may need to fix the firewall to allow multiple
     // ports to talk.  See http://www.kb.cert.org/vuls/id/800113

     // If your ISP provided one or more IP addresses for stable
     // nameservers, you probably want to use them as forwarders.
     // Uncomment the following block, and insert the addresses replacing
     // the all-0's placeholder.

     // forwarders {
     //   0.0.0.0;
     // };

     //========================================================================
     // If BIND logs error messages about the root key being expired,
     // you will need to update your keys.  See https://www.isc.org/bind-keys
     //========================================================================
     dnssec-enable no;
     //dnssec-validation auto;

     auth-nxdomain no;    # conform to RFC1035
     //listen-on-v6 { any; };

     listen-on port 53 { localhost; 10.0.1.0/24; };
     allow-query { localhost; 10.0.1.0/24; };

     recursion yes;
};
```

- Finalmente, usamos a funcionalidade BIND para confirmar que não há erros nos arquivos de configuração e reiniciamos o servidor BIND para aplicar as alterações na configuração

```
# named-checkcon

# service bind9 restart
```

Testando nosso servidor BIND de um cliente

Agora, para a parte de teste de nosso servidor recursivo, usaremos uma nova máquina virtual. Para fazer isso, rodamos a segunda máquina virtual (cliente) que instalamos anteriormente.

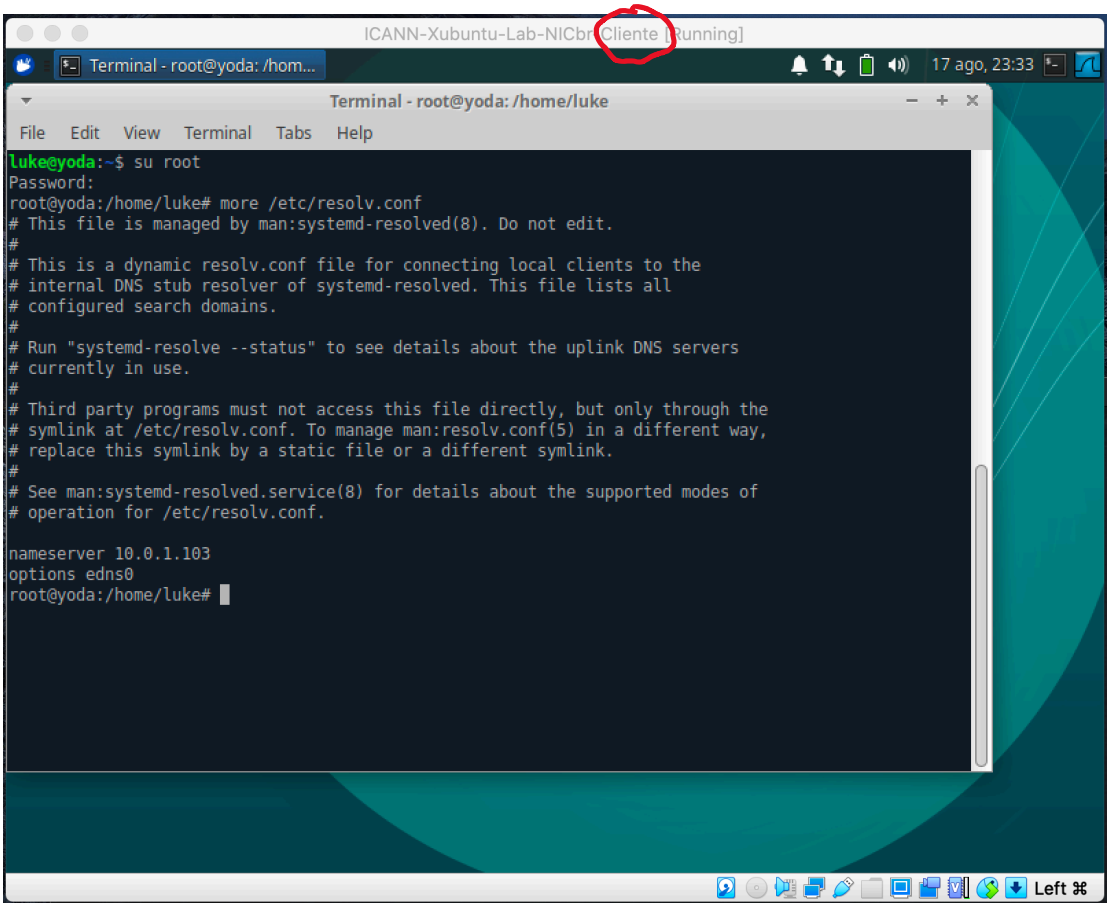


E ajustamos a configuração do cliente (no arquivo *resolv.conf*) para usar nosso servidor BIND

- Abrir o /etc/resolv.conf e incluir o endereço do servidor (usando usuário administrador)

Lembre-se de fazer esta configuração no CLIENTE e não no servidor !!!

```
$ su root        (Password: icannlab1)

# nano /etc/resolv.conf

nameserver 10.0.1.103
```



O comando DIG (de novo!)

Comando DIG

- ***Domain Information Groper***
  - Útil para troubleshooting e treinamentos
- Componente do pacote BIND (instalamos o bind-utils, lembra?)

```
$ dig [servidor] [nome] [tipo]
```

- Digite apenas dig no terminal

```
$ dig
```

- O que apareceu?
- Qual foi o tempo de resposta (Query time) ?

- Faça alguns testes com o dig

```
$ dig o-seu-site-favorito a
$ dig www.google.com a
$ dig lacnic.net mx
$ dig dominioquenaoexiste.algumtld any
$ dig afrinic.net aaaa
```

- Agora envie consultas pra outro resolver.

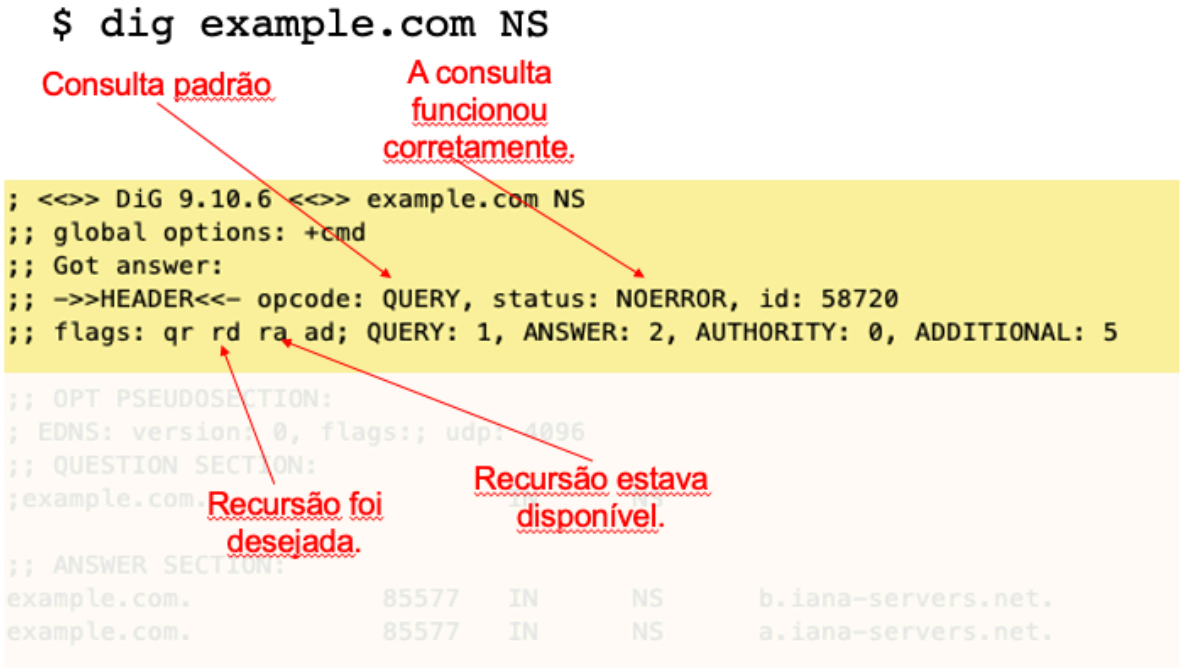
```
$ dig @8.8.8.8 o-seu-site-favorito a
$ dig @9.9.9.9 www.google.com a
```

Outros testes. Compare os tempos de resposta.

- Verificar o tempo de resposta dos servidores raiz

```
$ dig @l.root-servers.net id.server ch txt

$ dig @b.root-servers.net id.server ch txt
```

- Agora vamos decifrar um pouco mais a resposta do DIG.



- Teste de DNSSEC

```
$ dig dnssec-failed.org
```

O que acontece?

Implementando o DNSSEC no BIND

Modificamos o arquivo de configuração para ativar o DNSSEC

```
# nano named.conf.options

dnssec-enable yes;
dnssec-validation auto;
```

Usamos a funcionalidade BIND para confirmar que não há erros nos arquivos de configuração e reiniciamos o servidor BIND para aplicar as alterações na configuração

```
# named-checkcon                    (Se não retornar nada, está tudo bem)

# service bind9 restart
```

a. Teste

```
$ dig nic.br +dnssec +multi
```
(observe a flag ad: dado autenticado)

```
$ dig dnssec-failed.org
```
(Qual é o status?)

b. Verificando os novos Resource Records
DNSKEY
```
$ dig nic.br DNSKEY
```

DS
```
$ dig nic.br DS
```

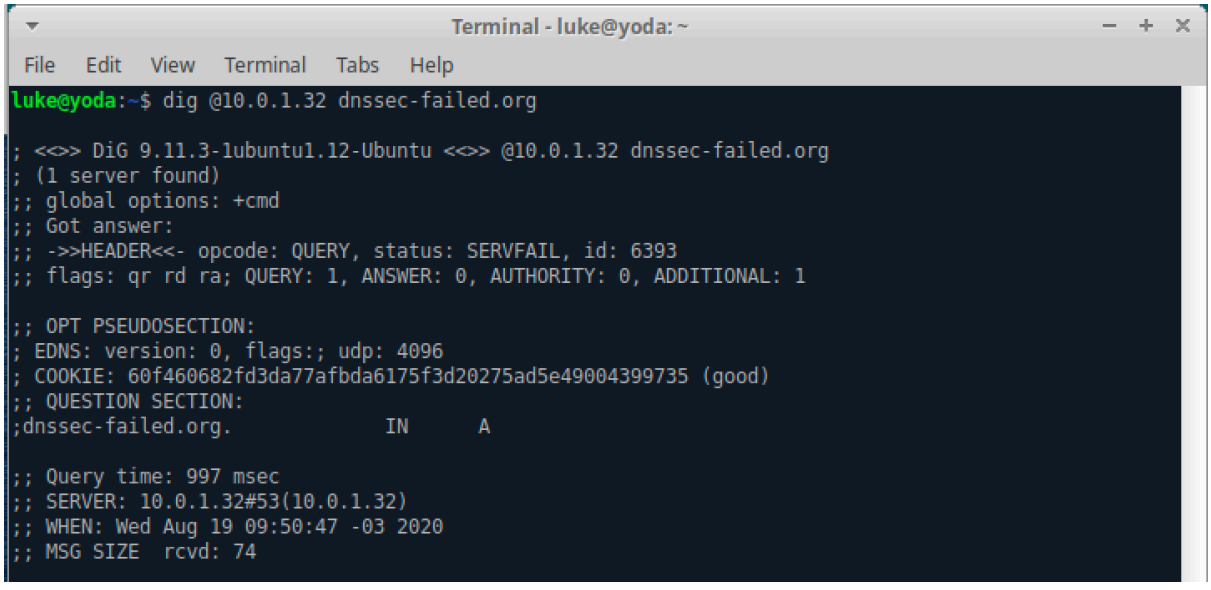
RRSIG
```
$ dig nic.br RRSIG
```

- Como criar uma exceção para o DNSSEC no BIND

As âncoras de confiança negativas (NTAs) podem ser usadas para mitigar as falhas de validação de DNSSEC desativando a validação de DNSSEC em domínios específicos.

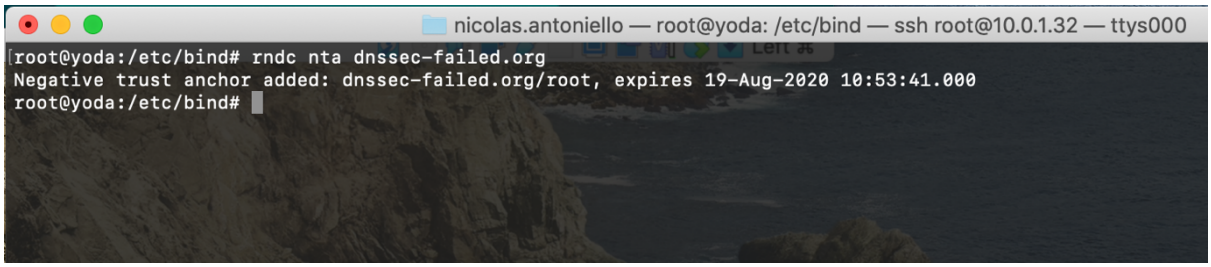
Por exemplo, usaremos o domínio *dnssec-failed.org* para testar este recurso.

Primeiro, a partir do cliente, fazemos uma consulta DNS para o domínio ***dnssec-failed.org***

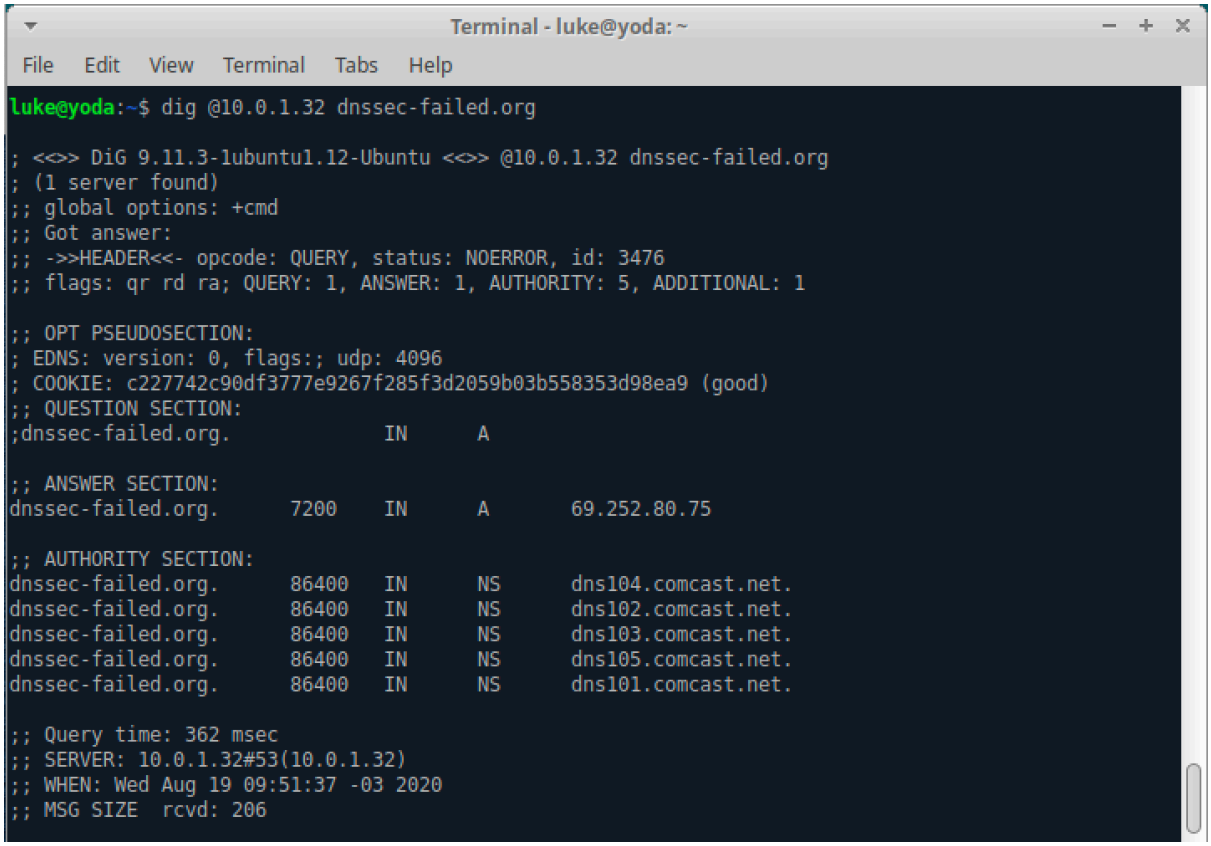


Agora vamos inserir a exceção no servidor DNS usando o comando:

```
# rndc nta dnssec-failed.org
```


nicolas.antoniello — root@yoda: /etc/bind — ssh root@10.0.1.32 — ttys000
root@yoda:/etc/bind# rndc nta dnssec-failed.org
Negative trust anchor added: dnssec-failed.org/root, expires 19-Aug-2020 10:53:41.000
root@yoda:/etc/bind#

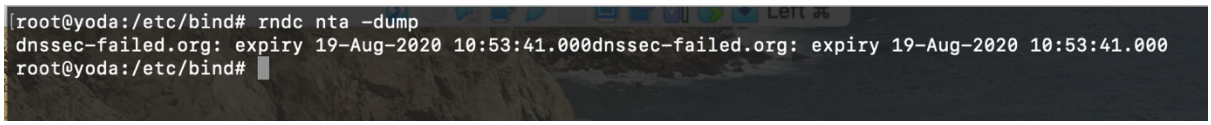
Por fim, fazemos a consulta DNS novamente do cliente para o mesmo domínio; agora obtendo a resposta sem validação DNSSEC.



O tempo de vida padrão de um NTA é de uma hora, embora, por padrão, o BIND faça uma pesquisa na zona a cada cinco minutos para verificar se a zona agora é validada corretamente, quando o NTA expirará automaticamente. O tempo de vida padrão e o intervalo de pesquisa podem ser configurados via *named.conf*, e o tempo de vida pode ser alterado por zona usando o parâmetro `-lifetime duration` no `rndc nta`. Ambos os valores do timer têm um valor máximo permitido de uma semana.

A lista de NTAs configurados pode ser visualizada com o comando:

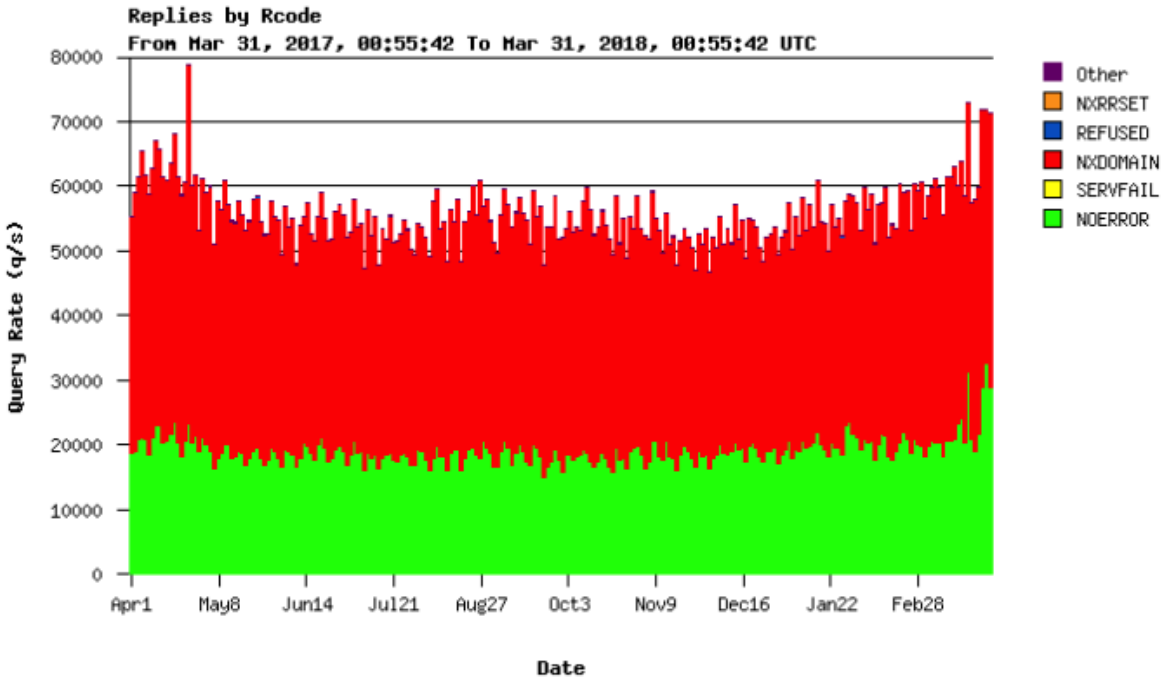
```
# rndc nta -dump
```



Hyperlocal implementado em nosso servidor DNS BIND

Da RFC7706:
“Alguns resolvedores recursivos DNS têm tempos de ida e volta mais longos do que o desejado para o servidor raiz DNS mais próximo. Alguns operadores de resolução recursiva de DNS desejam impedir a espionagem de solicitações enviadas a servidores raiz de DNS por terceiros. Esses resolvedores podem diminuir muito o tempo de ida e volta e evitar a observação de solicitações, executando uma cópia da zona raiz completa em um endereço de loopback (como 127.0.0.1). Este documento mostra como iniciar e manter uma cópia da zona raiz que não represente uma ameaça para outros usuários do DNS, ao custo de adicionar alguma fragilidade operacional para o operador.”

Curiosidade: 65% da carga dos servidores raiz é dedicada a NXDOMAIN



Yearly graph of return codes on K-Root, provided by the RIPE NCC

Fonte: https://www.ripe.net/analyse/dns/k-root/statistics/?type=ROOT&increment=yearly

Para configurar o ***hyperlocal***, faremos algumas alterações de configuração nos arquivos *named.conf* e *named.conf.options*. Para isso vamos gerar 2 views em BIND, uma para atender as consultas à zona Raiz (aquela que manterá a cópia dessa zona: hiperlocal) e a outra para atender as consultas recursivas como um servidor recursivo normalmente faz. O que acontecerá então é que o servidor recursivo em vez de consultar os servidores raiz para resolver o "." ele se consultará obtendo as informações

do banco de dados local *rootzone.db* (resultado da transferência da zona raiz com a configuração hyperlocal.

Teste antes da implementação

BIND recursivo sem Hyperlocal:

Observe que faremos um teste de consulta (vamos consultar do cliente e não do servidor) para um domínio de nível superior inexistente (NXDOMAIN) para verificar o tempo de resposta aproximado ao consultar um servidor raiz.

```
$ dig domaininvalid.com.br.xxxxxxx
```

E anotamos o tempo de resposta para essa consulta.

Editamos o arquivo de configuração *named.conf.options* da seguinte maneira:

```
# nano /etc/bind/named.conf.options
```

```
options {
        directory "/var/cache/bind";

        // If there is a firewall between you and nameservers you want
        // to talk to, you may need to fix the firewall to allow multiple
        // ports to talk.  See http://www.kb.cert.org/vuls/id/800113

        // If your ISP provided one or more IP addresses for stable
        // nameservers, you probably want to use them as forwarders.
        // Uncomment the following block, and insert the addresses replacing
        // the all-0's placeholder.

        // forwarders {
        //      8.8.8.8;
        // };

        //========================================================================
        // If BIND logs error messages about the root key being expired,
        // you will need to update your keys.  See https://www.isc.org/bind-keys
        //========================================================================
        dnssec-enable yes;
        dnssec-validation auto;

        auth-nxdomain no;    # conform to RFC1035
        //listen-on-v6 { any; };

        listen-on port 53 { localhost; 10.0.1.0/24; };
        allow-query { localhost; 10.0.1.0/24; };

        //recursion yes;
};
```

Editamos o arquivo de configuração *named.conf* da seguinte maneira:

```
# nano /etc/bind/named.conf
```

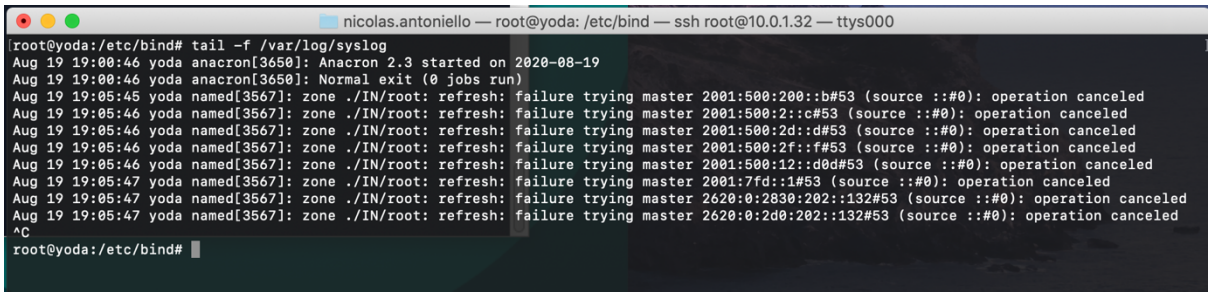
```
// This is the primary configuration file for the BIND DNS server named.
//
// Please read /usr/share/doc/bind9/README.Debian.gz for information on the
// structure of BIND configuration files in Debian, *BEFORE* you customize
// this configuration file.
//
// If you are just adding zones, please do that in /etc/bind/named.conf.local

include "/etc/bind/named.conf.options";
include "/etc/bind/named.conf.local";

// Comentamos o seguinte -include- porque nao usaremos esta configuracao.
// Em vez disso, usaremos a transferencia da zona raiz (hipelocal).
//include "/etc/bind/named.conf.default-zones";

// Criamos a visao para a zona raíz
view root {
        // Indicamos quais recursivos podem usar a copia local da raiz
        match-clients { localhost; };
        zone "." {
                type slave;
                file "rootzone.db"; // Este arquivo armazena copia da raiz
                notify no;
                masters {
                        199.9.14.201;           # b.root-servers.net
                        192.33.4.12;            # c.root-servers.net
                        199.7.91.13;            # d.root-servers.net
                        192.5.5.241;            # f.root-servers.net
                        192.112.36.4;           # g.root-servers.net
                        193.0.14.129;           # k.root-servers.net
                        192.0.47.132;           # xfr.cjr.dns.icann.org
                        192.0.32.132;           # xfr.lax.dns.icann.org
                        2001:500:200::b;        # b.root-servers.net
                        2001:500:2::c;          # c.root-servers.net
                        2001:500:2d::d;         # d.root-servers.net
                        2001:500:2f::f;         # f.root-servers.net
                        2001:500:12::d0d;       # g.root-servers.net
                        2001:7fd::1;            # k.root-servers.net
                        2620:0:2830:202::132;   # xfr.cjr.dns.icann.org
                        2620:0:2d0:202::132;    # xfr.lax.dns.icann.org
                };
        };
};

// Criamos a visao para atender consultas recursivas
view recursive {
        dnssec-enable yes;
        dnssec-validation auto;

        allow-recursion { localhost; 10.0.1.0/24; };
        recursion yes;

        // Para resolver a raiz usaremos a zona local (hyperlocal)
        zone "." {
                type static-stub;
```

```
        server-addresses { 127.0.0.1; };
};
    };
```

Usamos a funcionalidade BIND para confirmar que não há erros nos arquivos de configuração e reiniciamos o servidor BIND para aplicar as alterações na configuração

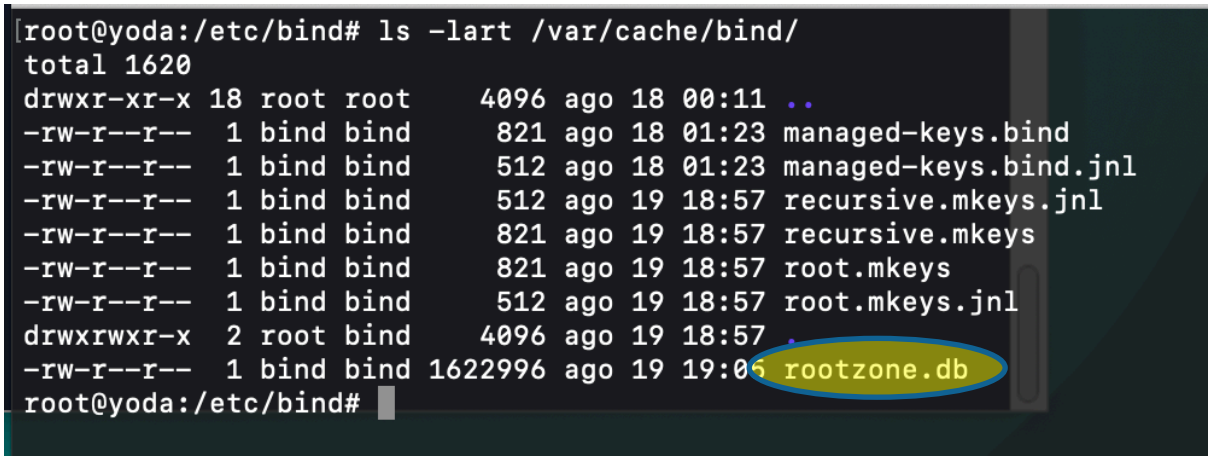
```
# named-checkconf      (Se não retornar nada, está tudo bem)
# service bind9 restart
```

Podemos verificar o log do sistema */var/log/syslog* para verificar se o BIND foi reiniciado corretamente (certamente teremos avisos de que os destinos IPv6 não estão acessíveis no caso de não termos conectividade IPv6).



Verificamos se o arquivo local com a cópia da zona raiz (*rootzone.db*) foi criado corretamente:

```
# ls -lart /var/cache/bind/
```



Teste da implementação

BIND recursivo com Hyperlocal

Vamos realizar alguns testes de consultas (vamos consultar do cliente e não do servidor) para domínios de nível superior inexistentes (NXDOMAIN) para verificar o tempo de resposta aproximado ao consultar diferentes servidores raiz.

Teste 1: Servidor público Google

```
$ dig @8.8.8.8 domaininvalid.com.br.xxxxxxx
```

Teste 2: Recursivo do meu provedor

```
$ dig @192.168.0.1 domaininvalid.com.br.xxxxxxx
```

Teste 3: Nosso servidor BIND recursivo com Hyperlocal

```
$ dig @10.0.1.32 domaininvalid.com.br.xxxxxxx
```

Comparamos o tempo obtido agora com o que obtivemos quando não tínhamos o Hyperlocal configurado.