



6° LABORATÓRIO DE REDES DE COMPUTADORES

Serviços Básicos de Rede – DNS

INTRODUÇÃO

Para o correto funcionamento de redes, alguns serviços de nível de aplicação complementam as funções básicas de rede, oferecendo funcionalidades que facilitam a administração da rede. O serviço DNS (*Domain Name Service*) permite consultas, de forma a se saber quais são os endereços de servidores, com base em seus nomes. Esta é a funcionalidade de resolução de nomes.

OBJETIVOS

1) Entender como funciona a implementação do DNS no Linux, e configura-la

REFERÊNCIAS TEÓRICAS

- Objetivo e funcionamento do protocolo DNS.
- O que é o *resolver*?
- Como são feitas as consultas ao serviço DNS?
- Software BIND (o mais usado atualmente para implementar um servidor DNS).

MATERIAL NECESSÁRIO

- Interfaces de rede (NIC's)
- Máquinas i386 com sistema GNU/Linux
- Cabos de rede – par trançado normal
- *Switches* ou HUBs
- Software nas máquinas: ambiente GNU/Linux básico, pacote DNS server (bind9)
- Acesso à Internet – NÃO é necessário

ROTEIRO

1) Montagem de rede interconectada para o experimento

Monte uma topologia de rede única com 3 ou mais máquinas no mesmo segmento. Configure a numeração de forma 192.168.10.* e teste com o comando ping.

2) Configuração estática local de mapeamentos endereços/nomes (através arquivo hosts)

Qualquer máquina com suporte TCP/IP possui um arquivo do tipo hosts (no UNIX/Linux: */etc/hosts*) onde estão configuradas associações fixas e estáticas entre endereços e nomes. Caso o nome sendo pedido não esteja neste arquivo, ele irá proceder a busca num servidor DNS.

Dê nomes às máquinas da rede montada que farão papel de estações nesta rede. Use o comando `hostname` para ver o nome atual de cada máquina e até, altera-lo. Note que este fica armazenado no arquivo `/etc/hostname`.

Teste a resolução de nomes local pelo arquivo `hosts` através do comando `ping`, executando um `ping` para um nome e não para o endereço.

Verifique qual é o domínio configurado no sistema com o comando `dnsdomainname`.

3) Configuração do cliente DNS (resolver)

Nos arquivos:

`/etc/resolv.conf`
`/etc/host.conf`

estão as configurações do cliente resolver DNS. Se a interface de rede em questão estiver configurada para DHCP, estas configurações serão, normalmente, automaticamente feitas pelo cliente DHCP da máquina. Caso contrário deve-se editar as configurações manualmente.

Usaremos o arquivo `resolv.conf`. Segue abaixo um exemplo comentado deste arquivo:

```
# Nome do domínio a ser usado para queries com nomes curtos
domain starwars.ucb.br

# entradas que dizem quais são os endereços dos servidores DNS
nameserver 192.168.10.1
nameserver 192.168.10.2

# Existem outra opções mais específicas (verificar a documentação)
```

Pronto! A estação está pronta para resolver nomes via DNS.

Partimos agora para a configuração dos servidores DNS da rede.

4) Configuração do Servidor DNS (bind)

Selecione algumas das máquinas para serem servidores DNS nesta rede e defina o seu domínio, de acordo com as próximas seções.

O arquivo de configuração original (`/etc/named.conf`) foi quebrado em vários arquivos que são incluídos por diretivas “include” no atual arquivo de configuração:

`/etc/bind/named.conf`

Note que os nomes e diretórios podem mudar de distribuição para distribuição. Os arquivos relacionados mais importantes neste pacote são:

`/etc/bind/named.conf` : contém configurações gerais do servidor DNS e zonas comuns.

`/etc/bind/named.conf.options` : contém opções gerais do servidor DNS.

`/etc/bind/named.conf.local` : contém configurações de zonas locais do servidor DNS.

`/var/cache/bind/` : diretório onde ficam as informações de cache. Este diretório pode ser alterado no arquivo `/etc/bind/named.conf.options`

As opções padrão nos arquivos de configuração gerais podem ser mantidas.

Dê uma olhada no arquivo de configurações gerais (`named.conf`). O que pode ser observado?

Iremos criar zonas adicionais usando o arquivo de configurações locais. Segue abaixo um exemplo comentado do arquivo `/etc/bind/named.conf.local` que conterá as zonas atendidas pelo servidor DNS:

```
# Cada zona é declarada como abaixo. os tipos são:
# primary-master: contém um cópia master dos dados da zona
# secondary-slave: replicas de uma master
# hint: aponta para os servidores root
# A opção "file" indica qual é o arquivo que contém os dados
# da zona "starwars" da classe IN (Internet)

zone "starwars.ucb.br" in {
    type master;
    file "/etc/bind/db.starwars";
};

# Zona para dns reverso
zone "10.168.192.in-addr.arpa" in {
    type master;
    file "/etc/bind/db.warsstar";
};
```

Uma vez definidas as zonas, deve-se criar os arquivos para cada uma das zonas master. Segue exemplo abaixo para o arquivo `db.starwars`:

```
# Cada definição de master deve se iniciar com uma entrada SOA
# Ela indica o servidor de nomes para o domínio em questão
# e parâmetros de operação

@      IN      SOA      vader.starwars.ucb.br. root.vader.starwars.ucb.br. (
2000092601 ;numero serial - deve ser incrementado a cada mudança neste
arquivo
21600      ;refresh      - das informações para slaves
1800      ;retry        - tempo entre as retentativas
604800    ;expire       - tempo para se desistir de contactar master
86400 )    ;mínimo      - tempo a manter a informação no cache (TTL)

starwars.ucb.br. IN      NS      vader.starwars.ucb.br.
starwars.ucb.br. IN      MX 10 R2D2.starwars.ucb.br      ;entrada MX (mail server)
localhost      IN      A      127.0.0.1
yoda           IN      A      192.168.10.1
obiwan        IN      A      192.168.10.2
leia          IN      A      192.168.10.3
luke          IN      A      192.168.10.4
vader         IN      A      192.168.10.66
R2D2          IN      A      192.168.10.100
```

Se possível, crie também um arquivo para o DNS reverso (arquivo `db.warsstar`)

```
# Realiza a resolução reversa
# O tipo PTR significa um alias para o endereço IP

@      IN      SOA      vader.starwars.ucb.br. root.vader.starwars.ucb.br. (
2000092601
21600
1800
604800
86400 )

      IN      NS      vader.starwars.ucb.br.
1     IN      PTR     yoda.starwars.ucb.br
```

2	IN	PTR	obiwan.starwars.ucb.br
3	IN	PTR	leia.starwars.ucb.br
4	IN	PTR	luke.starwars.ucb.br
66	IN	PTR	vader.starwars.ucb.br
100	IN	PTR	R2D2.starwars.ucb.br

Lembrar os endereços IP dos servidores DNS configurados anteriormente no cliente.

Após a edição do arquivo, deve-se reiniciar o servidor para que as alterações tenham efeito.

```
/etc/init.d/bind9 start
```

esta é a maneira correta de disparar serviços num servidor Linux, porém queremos ver o que está acontecendo com o servidor na sua tela. Por isso vamos disparar o servidor “na mão” com o comando:

```
/usr/sbin/named -f -g -d 1
```

as opções acima mostram as mensagens de *debug* na tela e não dão o “fork” no processo. Para encerrar o processo use o comando `kill` ou `killall` (dar CTRL-Z apenas o manda para o background!).

Teste a resolução de nomes com o comando: `host -d <nome>`

e teste a resolução inversa com o comando: `host -d <ip>`

Outro comando que pode ser usado para testar a resolução é o comando `nslookup`. Faça consultas, por exemplo:

```
> luke.starwars.ucb.br
```

Faça consultas usando outro servidor:

```
> server <nome ou IP>
```

observe se as respostas são autoritativas ou não. A opção “`set=_`” muda o tipo de registro a ser pedido do servidor DNS. Faça consultas reversas:

```
> set type=ptr
> 192.168.0.4
```

Consulte os servidores de mail (MX) para o domínio

```
> set type=mx
> ucb.br
```

Todas estas opções existem também para o comando `host`.

QUESTÕES PARA O RELATÓRIO

- 1) Qual é o retorno do comando `dnsdomainname`? O que significa?
- 2) O que é o nome *localhost*? E o endereço 127.0.0.1 dado a ele no arquivo `/etc/hosts`? Por que deve sempre existir este endereço e nome em sistemas UNIX/Linux?
- 3) O que é o FQDN?
- 4) Podemos ter 2 servidores DNS na mesma rede? Qual é a configuração mais adequada para esta situação?
- 5) O que é DNS reverso? Como isto foi implementado no lab?
- 6) O que é a entrada MX inserida no domínio? Podem haver mais de uma?
- 7) O que é resposta autoritativa dada por um servidor DNS? Explique.

8) O que é um servidor *caching-only*?