

```
root@ssh:/# ps aux | grep usr/sbin/sshd
root      220  0.0  0.0  69960  5532 ?        Ss   13:54   0:00 /usr/sbin/ssh
d
root      451  0.0  0.0  11116   924 pts/5    S+   14:10   0:00 grep usr/sbin
/ssh
```

2. Agora vamos testar a conectividade do serviço fazendo uma acesso remoto, por exemplo, no terminal do pc2 execute: **ssh 10.0.9.10**.

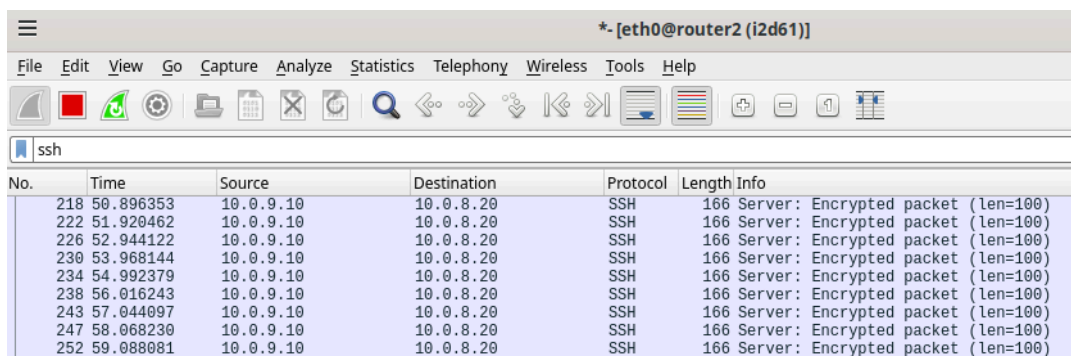
Observe e salve que o prompt do seu terminal mudou para **root@ssh:~#**, isso significa que, apesar de você estar no terminal do pc2, você está conectado no SSH SERVER. Tudo que você digitar estará sendo executado no SSH SERVER.

```
root@pc2:~# ssh 10.0.9.10
root@10.0.9.10's password:
Linux ssh 6.1.0-26-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.112-1 (2024-09-30) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

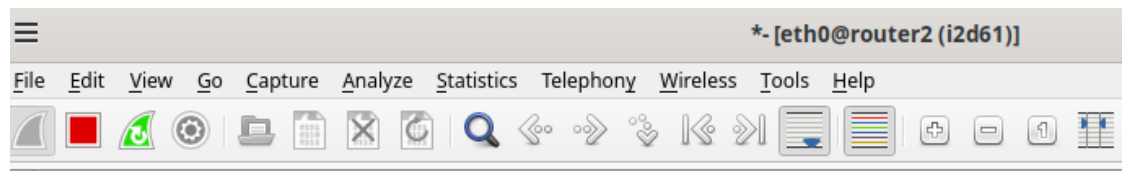
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Oct 21 14:10:03 2024 from 10.0.8.20
root@ssh:~#
```

3. Agora vamos capturar pacotes do ssh. Basta usar o Wireshark em qualquer interface onde passam os pacotes. Por exemplo, no router2. Recorte a tela do Wireshark, filtrando os pacotes do ssh. Mostre o encapsulamento de pacotes de aplicação e seu posicionamento na estrutura de pacotes.



No.	Time	Source	Destination	Protocol	Length	Info
218	50.896353	10.0.9.10	10.0.8.20	SSH	166	Server: Encrypted packet (len=100)
222	51.920462	10.0.9.10	10.0.8.20	SSH	166	Server: Encrypted packet (len=100)
226	52.944122	10.0.9.10	10.0.8.20	SSH	166	Server: Encrypted packet (len=100)
230	53.968144	10.0.9.10	10.0.8.20	SSH	166	Server: Encrypted packet (len=100)
234	54.992379	10.0.9.10	10.0.8.20	SSH	166	Server: Encrypted packet (len=100)
238	56.016243	10.0.9.10	10.0.8.20	SSH	166	Server: Encrypted packet (len=100)
243	57.044097	10.0.9.10	10.0.8.20	SSH	166	Server: Encrypted packet (len=100)
247	58.068230	10.0.9.10	10.0.8.20	SSH	166	Server: Encrypted packet (len=100)
252	59.088081	10.0.9.10	10.0.8.20	SSH	166	Server: Encrypted packet (len=100)

4. Recorte a tela do Wireshark, filtrando os pacotes do icmp. Comprovando que os pacotes do ping estão passando pelo router2.



No.	Time	Source	Destination	Protocol	Length	Info
126	28.368017	10.0.8.20	10.0.9.10	ICMP	98	Echo (ping) reply
129	29.391872	10.0.9.10	10.0.8.20	ICMP	98	Echo (ping) request
130	29.391924	10.0.8.20	10.0.9.10	ICMP	98	Echo (ping) reply
134	30.415849	10.0.9.10	10.0.8.20	ICMP	98	Echo (ping) request
135	30.415904	10.0.8.20	10.0.9.10	ICMP	98	Echo (ping) reply
138	31.439973	10.0.9.10	10.0.8.20	ICMP	98	Echo (ping) request
139	31.440049	10.0.8.20	10.0.9.10	ICMP	98	Echo (ping) reply
142	32.463992	10.0.9.10	10.0.8.20	ICMP	98	Echo (ping) request
143	32.464066	10.0.8.20	10.0.9.10	ICMP	98	Echo (ping) reply

5. Para encerrar a conexão ao SSH SERVER, no terminal do pc2 digite: exit. Observe e salve que o prompt do seu terminal mudou para **root@pc2:~#**, isso significa que a conexão foi encerrada.

```
root@ssh:~# exit
logout
Connection to 10.0.9.10 closed.
```

Servidor DNS

“Vai permitir a navegação através de nomes de máquinas. Com este serviço vamos configurar o banco de dados de um servidor DNS, BIND. Ele dará suporte a navegação por nomes de máquinas”

1. Criando um domínio de Internet

Apenas documentando os passos: Foi aberto o terminal do DNS_Server, acessado o diretório bind `cd /etc/bind` e definida uma zona (redes.edu.br) ao editar um arquivo `nano named.conf.default-zones` e inserir o seguinte conteúdo: `zone "redes.edu.br" { type master; file "/etc/bind/db.redes"; };`

2. Criando a base de dados relativa ao domínio

Apenas documentando os passos: Atribuir endereços IPv4 as máquinas db.redes ao configurar: `nano db.redes` com o seguinte conteúdo:

```

$TTL 86400
@ IN SOA ns.redes.edu.br. root (
    2022051200 ; Serial
    604800 ; Refresh
    86400 ; Retry
    2419200 ; Expire
    86400 ) ; Negative Cache TTL
;
@ IN NS ns.redes.edu.br.
@ IN MX 10 mail.redes.edu.br.
$ORIGIN redes.edu.br.
ns IN A 10.0.6.10
www IN A 10.0.9.11
ssh IN A 10.0.9.10
mail IN A 10.0.6.10
apelido IN CNAME mail.redes.edu.br.

```

e ao final, reiniciar o serviço DNS: `/etc/init.d/bind9 restart`

3. Faça um teste com consulta ao seu servidor com o comando, por exemplo `dig @localhost www.redes.edu.br`

```

root@DNS_Server:/etc/bind# dig @localhost www.redes.edu.br

; <<>> DiG 9.10.3-P4-Debian <<>> @localhost www.redes.edu.br
; (2 servers found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 35925
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.redes.edu.br. IN A

;; ANSWER SECTION:
www.redes.edu.br. 86400 IN A 10.0.9.11

```

4. Configurando as máquinas para acessarem o DNS

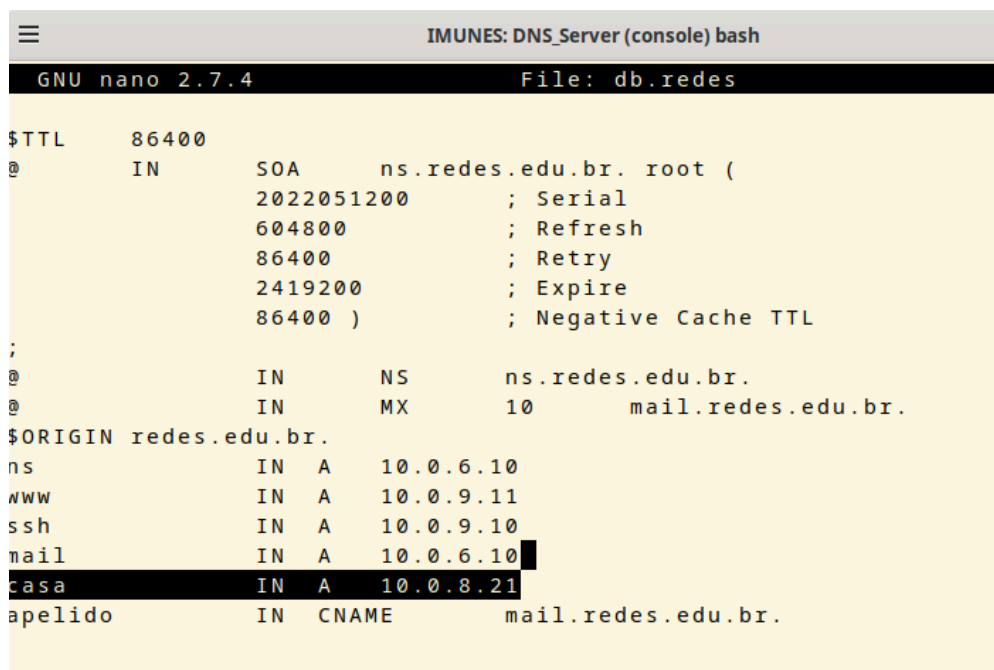
Em qualquer máquina que desejar navegar por nomes, declare o servidor DNS_Server como servidor DNS com o seguinte comando digitado no respectivo terminal: `echo nameserver 10.0.6.10 >> /etc/resolv.conf`

5. Faça alguns testes simples via ping como, por exemplo: ping www.redes.edu.br

```
root@DNS_Server:/etc/bind# ping www.redes.edu.br
PING www.redes.edu.br (10.0.9.11) 56(84) bytes of data.
64 bytes from 10.0.9.11 (10.0.9.11): icmp_seq=1 ttl=60 time=0.149 ms
64 bytes from 10.0.9.11 (10.0.9.11): icmp_seq=2 ttl=60 time=0.194 ms
64 bytes from 10.0.9.11 (10.0.9.11): icmp_seq=3 ttl=60 time=0.193 ms
^C
--- www.redes.edu.br ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2029ms
rtt min/avg/max/mdev = 0.149/0.178/0.194/0.026 ms
```

6. Acrescente um novo endereço ao banco de dados: casa.redes.edu.br apontando para o IP 10.0.8.21.

Foi alterado o arquivo db.redes `nano /etc/bind/db.redes` e adicionada a linha destacada abaixo:



```
IMUNES: DNS_Server (console) bash
GNU nano 2.7.4 File: db.redes

$TTL      86400
@          IN      SOA      ns.redes.edu.br. root (
                2022051200      ; Serial
                604800          ; Refresh
                86400           ; Retry
                2419200         ; Expire
                86400 )         ; Negative Cache TTL
;
@          IN      NS       ns.redes.edu.br.
@          IN      MX       10      mail.redes.edu.br.
$ORIGIN redes.edu.br.
ns         IN      A        10.0.6.10
www        IN      A        10.0.9.11
ssh        IN      A        10.0.9.10
mail       IN      A        10.0.6.10
casa      IN      A        10.0.8.21
apelido    IN      CNAME     mail.redes.edu.br.
```

Além disso, foi executado o seguinte comando: `echo "nameserver 10.0.6.10" >> /etc/resolv.conf` para declarar o DNS_Server como servidor DNS.

7. Prove que o mesmo está funcional, por exemplo, dando um ping casa.redes.edu.br a partir do host ssh.

```
root@ssh:~# ping casa.redes.edu.br
PING casa.redes.edu.br (10.0.8.21) 56(84) bytes of data.
64 bytes from 10.0.8.21 (10.0.8.21): icmp_seq=1 ttl=60 time=0.736 ms
64 bytes from 10.0.8.21 (10.0.8.21): icmp_seq=2 ttl=60 time=0.169 ms
64 bytes from 10.0.8.21 (10.0.8.21): icmp_seq=3 ttl=60 time=0.218 ms
64 bytes from 10.0.8.21 (10.0.8.21): icmp_seq=4 ttl=60 time=0.195 ms
64 bytes from 10.0.8.21 (10.0.8.21): icmp_seq=5 ttl=60 time=0.194 ms
64 bytes from 10.0.8.21 (10.0.8.21): icmp_seq=6 ttl=60 time=0.191 ms
^C
--- casa.redes.edu.br ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5105ms
rtt min/avg/max/mdev = 0.169/0.283/0.736/0.203 ms
```

8. No terminal do DNS_Server execute: dig apelido.redes.edu.br

- Qual foi o resultado obtido?
- Qual o significado?

```
IMUNES: DNS_Server (console) bash

<<>> DiG 9.10.3-P4-Debian <<>> apelido.redes.edu.br
; global options: +cmd
; Got answer:
; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 17308
; flags: qr aa rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 1, ADDITIONAL: 2

; OPT PSEUDOSECTION:
EDNS: version: 0, flags:; udp: 4096
; QUESTION SECTION:
apelido.redes.edu.br.          IN      A

; ANSWER SECTION:
apelido.redes.edu.br.      86400   IN      CNAME   mail.redes.edu.br.
mail.redes.edu.br.        86400   IN      A       10.0.6.10

; AUTHORITY SECTION:
redes.edu.br.              86400   IN      NS       ns.redes.edu.br.

; ADDITIONAL SECTION:
ns.redes.edu.br.           86400   IN      A       10.0.6.10

; Query time: 0 msec
; SERVER: 10.0.6.10#53(10.0.6.10)
; WHEN: Mon Oct 21 15:15:20 UTC 2024
; MSG SIZE rcvd: 117
```

O DNS realizou uma consulta com o seguinte resultado: 1 consulta, 2 respostas, 1 autoridade e 1 informação adicional. Essas seções demonstram que o DNS resolveu corretamente o domínio apelido.redes.edu.br para o IP 10.0.6.10.

Servidor WEB

“Permite hospedar e acessar remotamente páginas da Internet”

1. Preparando uma página HTML para colocar no servidor WEB. No terminal WEB Server, `cd /var/www/html` e utilize o editor nano para editar uma página chamada `index.html` `nano index.html`. Salve um print do conteúdo. Ao final, inicie o serviço WEB `/etc/init.d/lighttpd start`

```
root@web:/var/www/html# cat index.html
<html>
<body>
<h1>Redes de Computadores</h1>
<p>Pagina teste da aluna Luiza Kuze</p>
</body>
</html>
```

2. Faça um acesso a sua página, a partir do Firefox (cliente HTTP) em um PC cliente de sua escolha:

Escolhi acesso pelo pc1, o que resultou na imagem abaixo:

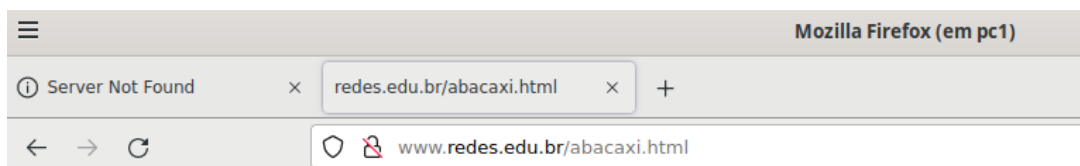


3. Salve um print da tela do Wireshark destacando a troca de mensagens HTTP e o conteúdo HTML de uma das páginas acessadas.

The screenshot shows the Wireshark interface with a packet capture on the 'eth0' interface. The packet list shows several HTTP messages. The selected packet is packet 104, which is an HTTP GET request for '/ HTTP/1.1' from source 10.0.8.21 to destination 10.0.9.11. The packet details pane shows the structure of the HTTP request, including the method (GET), URI (/), and version (HTTP/1.1).

No.	Time	Source	Destination	Protocol	Length	Info
104	508.442115	10.0.8.21	10.0.9.11	HTTP	393	GET / HTTP/1.1
106	508.443070	10.0.9.11	10.0.8.21	HTTP	401	HTTP/1.1 200 OK (text/html)
108	508.540246	10.0.8.21	10.0.9.11	HTTP	349	GET /favicon.ico HTTP/1.1
109	508.540412	10.0.9.11	10.0.8.21	HTTP/X	545	HTTP/1.1 404 Not Found
130	584.395966	10.0.8.21	10.0.9.11	HTTP	498	GET / HTTP/1.1
132	584.396436	10.0.9.11	10.0.8.21	HTTP	382	HTTP/1.1 200 OK (text/html)

4. . Crie uma nova página dentro do diretório /var/www/html: **nano abacaxi.html**. Recorte a tela com a página em destaque no navegador e cole no relatório. Salve um print da tela do Wireshark destacando a troca de mensagens HTTP e o conteúdo HTML de uma das páginas acessadas.



Redes de Computadores

Minha primeira pagina.....

The screenshot shows the Wireshark interface with a capture of network traffic. The filter is set to 'http'. The packet list shows four packets related to the HTTP request and response for the URL /abacaxi.html.

No.	Time	Source	Destination	Protocol	Length	Info
114	622.399281	10.0.8.21	10.0.9.11	HTTP	405	GET /abacaxi.html HTTP/1.1
116	622.402496	10.0.9.11	10.0.8.21	HTTP	381	HTTP/1.1 200 OK (text/html)
118	622.501692	10.0.8.21	10.0.9.11	HTTP	361	GET /favicon.ico HTTP/1.1
119	622.501854	10.0.9.11	10.0.8.21	HTTP/X...	545	HTTP/1.1 404 Not Found