

Comunicações por Computador

Trabalho Prático 3: Serviço de Resolução de Nomes (DNS)

Ana Rita Peixoto, Leonardo Marreiros, and Luís Pinto

University of Minho, Department of Informatics, 4710-057 Braga, Portugal

e-mail: {a89612,a89537,a89506}@alunos.uminho.pt

Parte 1: Consultas ao serviço de nomes DNS

Alínea a. Qual o conteúdo do ficheiro `/etc/resolv.conf` e para que serve essa informação?

O ficheiro `/etc/resolv.conf` define como o sistema utiliza o DNS para determinar os *host names* e endereços IP. Este ficheiro contém uma linha que especifica os domínios de procura e até 3 linhas que especificam os endereços IP do servidor DNS.

Além disso, pela consulta do manual do *nslookup* podemos verificar que este ficheiro está diretamente relacionado com o DNS.

```
core@xubuncore:~$ cat /etc/resolv.conf
# This file is managed by man:systemd-resolved(8). Do not edit.
#
# This is a dynamic resolv.conf file for connecting local clients to the
# internal DNS stub resolver of systemd-resolved. This file lists all
# configured search domains.
#
# Run "systemd-resolve --status" to see details about the uplink DNS servers
# currently in use.
#
# Third party programs must not access this file directly, but only through the
# symlink at /etc/resolv.conf. To manage man:resolv.conf(5) in a different way,
# replace this symlink by a static file or a different symlink.
#
# See man:systemd-resolved.service(8) for details about the supported modes of
# operation for /etc/resolv.conf.

nameserver 127.0.0.53
options edns0
search eduroam.uminho.pt
```

Fig. 1: Conteúdo do ficheiro

```
FILES
  /etc/resolv.conf
```

Fig. 2: Manual *Nslookup*

Alínea b. Os servidores `www.uminho.pt` e `www.ubuntu.com` têm endereços IPv6? Se sim, quais?

Neste procedimento, usamos o comando `Nslookup` para exibir as informações do serviço de nomes IPv6 onde especificamos o *resource record* como `AAAA`.

Como podemos verificar pela Figura 3, o servidor "`www.uminho.pt`" não possui endereços IPv6 enquanto que o servidor "`www.ubuntu.com`" possui os seguintes endereços IPv6:

- 2001:67c:1360:8001::2c
- 2001:67c:1360:8001::2b

```
> set q=aaaa
> www.uminho.pt
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
*** Can't find www.uminho.pt: No answer
> www.ubuntu.com
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
Name:   www.ubuntu.com
Address: 2001:67c:1360:8001::2c
Name:   www.ubuntu.com
Address: 2001:67c:1360:8001::2b
```

Fig. 3: Endereços IPv6 dos servidores

Alínea c. Quais os servidores de nomes definidos para os domínios: "`sapo.pt`", "`pt.`" e "`.`"?

De modo a verificar os servidores de nomes para os domínios especificados, foi necessário utilizar a *query NS (name server)* do `nslookup`. Embora tenhamos obtido uma resposta não autoritativa, podemos observar os resultados obtidos na seguinte figura, que contém os nomes relativos a cada domínio.

```
> set q=NS
> sapo.pt.
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
sapo.pt nameserver = ns.sapo.pt.
sapo.pt nameserver = dns02.sapo.pt.
sapo.pt nameserver = dns01.sapo.pt.
sapo.pt nameserver = ns2.sapo.pt.

Authoritative answers can be found from:
> pt.
```

(a)

```
> pt.
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
pt nameserver = b.dns.pt.
pt nameserver = c.dns.pt.
pt nameserver = ns.dns.br.
pt nameserver = d.dns.pt.
pt nameserver = a.dns.pt.
pt nameserver = e.dns.pt.
pt nameserver = h.dns.pt.
pt nameserver = ns2.nic.fr.
pt nameserver = g.dns.pt.

Authoritative answers can be found from:
> .
```

(b)

```
> .
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
nameserver = i.root-servers.net.
nameserver = c.root-servers.net.
nameserver = h.root-servers.net.
nameserver = f.root-servers.net.
nameserver = k.root-servers.net.
nameserver = g.root-servers.net.
nameserver = e.root-servers.net.
nameserver = m.root-servers.net.
nameserver = a.root-servers.net.
nameserver = j.root-servers.net.
nameserver = l.root-servers.net.
nameserver = b.root-servers.net.
nameserver = d.root-servers.net.

Authoritative answers can be found from:
>
```

(c)

Fig. 4: Servidores de nomes definidos para os domínios: (a) "`sapo.pt.`"; (b) "`pt.`"; (c) "`.`".

Alínea d. Existe o domínio open.money.? Será que open.money. é um host ou um domínio?

Como podemos ver pela imagem a seguir, o domínio open.money., de facto, existe. Para além disso, trata-se de um host uma vez que possui endereço IP.

```
core@core-VirtualBox:~$ host open.money.
open.money has address 35.154.208.116
open.money mail is handled by 5 alt2.aspmx.l.google.com.
open.money mail is handled by 10 mailstore1.secureserver.net.
open.money mail is handled by 10 alt4.aspmx.l.google.com.
open.money mail is handled by 1 aspmx.l.google.com.
open.money mail is handled by 5 alt1.aspmx.l.google.com.
open.money mail is handled by 10 alt3.aspmx.l.google.com.
open.money mail is handled by 0 smtp.secureserver.net.
core@core-VirtualBox:~$
```

Fig. 5: Domínio open.money

Alínea e. Qual é o servidor DNS primário definido para o domínio un.org.? Este servidor primário (master) aceita queries recursivas? Porquê?

O servidor DNS primário para o domínio un.org. é o servidor ns1.un.org, tal como é possível observar na seguinte figura. Concluímos que não é recursivo porque não conseguiu encontrar o servidor www.uminho.pt.

```
core@xubuncore:~$ nslookup -querytype=soa un.org.
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
un.org
  origin = ns1.un.org
  mail addr = root.un.org
  serial = 2021042400
  refresh = 1200
  retry = 3600
  expire = 1209600
  minimum = 300

Authoritative answers can be found from:

core@xubuncore:~$ ^C
core@xubuncore:~$ nslookup
> server ns1.un.org
Default server: ns1.un.org
Address: 157.150.185.28#53
> www.uminho.pt
Server:      ns1.un.org
Address:     157.150.185.28#53

** server can't find www.uminho.pt: REFUSED
>
```

Fig. 6: Servidor DNS primário do domínio un.org.

Alínea f. Obtenha uma resposta “autoritativa” para a questão anterior.

Para obter uma resposta autoritativa é necessário ter em conta o servidor primário (ns1.un.org). Conseguiamos obter esta informação a partir da alínea anterior. Após alterar o servidor *default* para o servidor primário, podemos efetuar o *nslookup* para o endereço un.org e obter a resposta autoritativa, tal como é visível na figura seguinte.

```
core@xubuncore:~$ nslookup
> server ns1.un.org
Default server: ns1.un.org
Address: 157.150.185.28#53
> un.org.
Server:      ns1.un.org
Address:     157.150.185.28#53

Name:   un.org
Address: 157.150.185.49
```

Fig. 7: Resposta autoritativa

Alínea g. Onde são entregues as mensagens de correio eletrónico dirigidas a presidency@eu.eu ou presidencia@2021portugal.eu?

De modo a verificar o local de entrega das mensagens de correio eletrónico, foi necessário utilizar a query MX (*mail exchange record*) do *nslookup*. Tal como é possível observar nas imagens abaixo, as mensagens dirigidas a presidency@eu.eu são entregues em smtp02.level27.be e em smtp01.level27.be. No entanto, o endereço smtp01.level27.be. é o principal e possui maior prioridade. As mensagens para o endereço presidencia@2021portugal.eu são entregues em mxg.eu.mpssec.net.

```
core@xubuncore:~$ nslookup
> set q=mx
> eu.eu
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
eu.eu  mail exchanger = 20 smtp02.level27.be.
eu.eu  mail exchanger = 10 smtp01.level27.be.
```

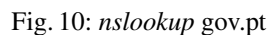
Fig. 8: Mensagens dirigidas a presidency@eu.eu

```
> 2021portugal.eu
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
2021portugal.eu mail exchanger = 10 mxg.eu.mpssec.net.
```

Fig. 9: Mensagens dirigidas a presidencia@2021portugal.eu

Para aceder a todas as informações acerca de gov.pt, através de DNS, optamos por utilizar o *nslookup* em conjunção com a definição " set q=any ". Os valores obtidos encontram-se inframencionados.



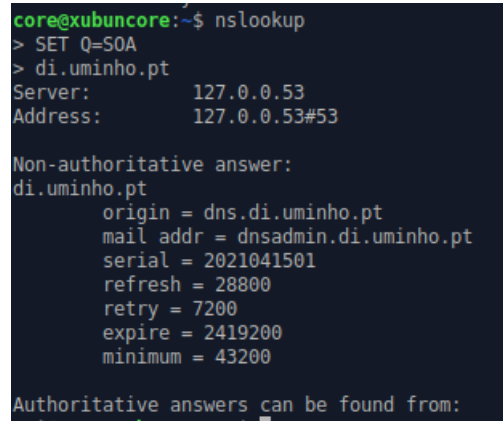
```
gov.pt origin = dnssec.gov.pt  
gov.pt mail addr = dns.ceger.gov.pt  
gov.pt serial = 2019071837  
gov.pt refresh = 18000  
gov.pt retry = 7200  
gov.pt expire = 2419200  
gov.pt minimum = 86400  
gov.pt rdata_46 = DNSKEY 10 3 86400 2
```

```
gov.pt rdata_43 = 51381 10 2 380408804504F3  
gov.pt nameserver = europel.dnsnode.net.  
gov.pt nameserver = ns02.fccn.pt.  
gov.pt nameserver = dns1.gov.pt.  
gov.pt nameserver = nsp.dnsnode.net.  
gov.pt nameserver = a.dns.pt.
```

```
Authoritative answers can be found from:  
>
```

Fig. 11: Datos relevantes -(a) "SOA record"; (b) "nameserver"

Após interrogar o *nslookup* quanto ao endereço IPv6 em questão, conseguimos obter informações como o nome do servidor (smtp01.fccn.pt). Para isso, foi necessário utilizar a *query ptr*. De modo a descobrir o contacto em caso de problemas, foi necessário utilizar a *query soa* e questionar o *nslookup* quanto ao domínio fccn.pt. Tal como podemos observar na seguinte figura, o contacto é o hostmaster.fccn.pt.

A terminal window with a dark background and light green text. The prompt is 'core@xubuncore:~\$'. The user enters 'nslookup'. The prompt changes to '>'. The user enters 'SET Q=SOA'. The prompt changes to '>'. The user enters 'di.uminho.pt'. The output shows 'Server: 127.0.0.53' and 'Address: 127.0.0.53#53'. Then 'Non-authoritative answer:' is shown, followed by 'di.uminho.pt' and a list of parameters: 'origin = dns.di.uminho.pt', 'mail addr = dnsadmin.di.uminho.pt', 'serial = 2021041501', 'refresh = 28800', 'retry = 7200', 'expire = 2419200', and 'minimum = 43200'. At the bottom, it says 'Authoritative answers can be found from:'.

```
core@xubuncore:~$ nslookup
> SET Q=SOA
> di.uminho.pt
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
di.uminho.pt
      origin = dns.di.uminho.pt
      mail addr = dnsadmin.di.uminho.pt
      serial = 2021041501
      refresh = 28800
      retry = 7200
      expire = 2419200
      minimum = 43200

Authoritative answers can be found from:
```

Fig. 14: Conteúdo do ficheiro

Cada domínio deve ter um registo de Início de Autoridade (*Start of Authority*) no ponto de transição onde o domínio é delegado do seu domínio pai. Um registo SOA inclui os seguintes detalhes relevantes à transferência de zona:

- *origin*: corresponde ao DNS primário definido para o domínio.
- *serial*: corresponde a um *timestamp* que é atualizado sempre que o domínio muda.
- *refresh*: corresponde ao número de segundos antes que a zona seja atualizada.
- *retry*: corresponde ao número de segundos até que uma atualização com falha deve ser tentada novamente.
- *expire*: corresponde ao limite superior em segundos antes de uma zona ser considerada não autoritativa.

Uma forma simplificada de como este mecanismo resulta é o seguinte: um domínio é constituído por um servidor DNS primário onde está localizada a base de dados, e servidores secundários. Estes servidores secundários utilizam o servidor primário para aceder à base de dados. Quando o secundário pretende aceder ao primário os seus *serials* são comparados (de lembrar que o *serial* do servidor primário é atualizado a cada *refresh* segundos ou sempre que a base de dados é alterada). Caso estes números sejam diferentes, isto significa que o secundário se encontra desatualizado e há uma falha, não ocorrendo a transferência. Sempre que ocorre esta falha, é feita novamente esta comparação a cada *retry* segundos. Finalmente, se passado *expire* segundos os *serials* continuem diferentes, isto significa que o secundário se encontra muito desatualizado quando comparado com o primário e este servidor secundário deixa então de responder a *queries*.

Parte 2: Instalação, configuração e teste de um domínio CC.PT

Nesta parte 2 do trabalho prático 3 foi proposta a criação de um domínio CC.PT para a topologia de rede da figura abaixo, de forma a poder usar nomes em vez de endereços IP. De forma a concretizar estes objetivos, seguiram-se os passos enunciados no guião. De seguida apresentam-se os testes efetuados que comprovam o funcionamento dos servidores primário e secundário.

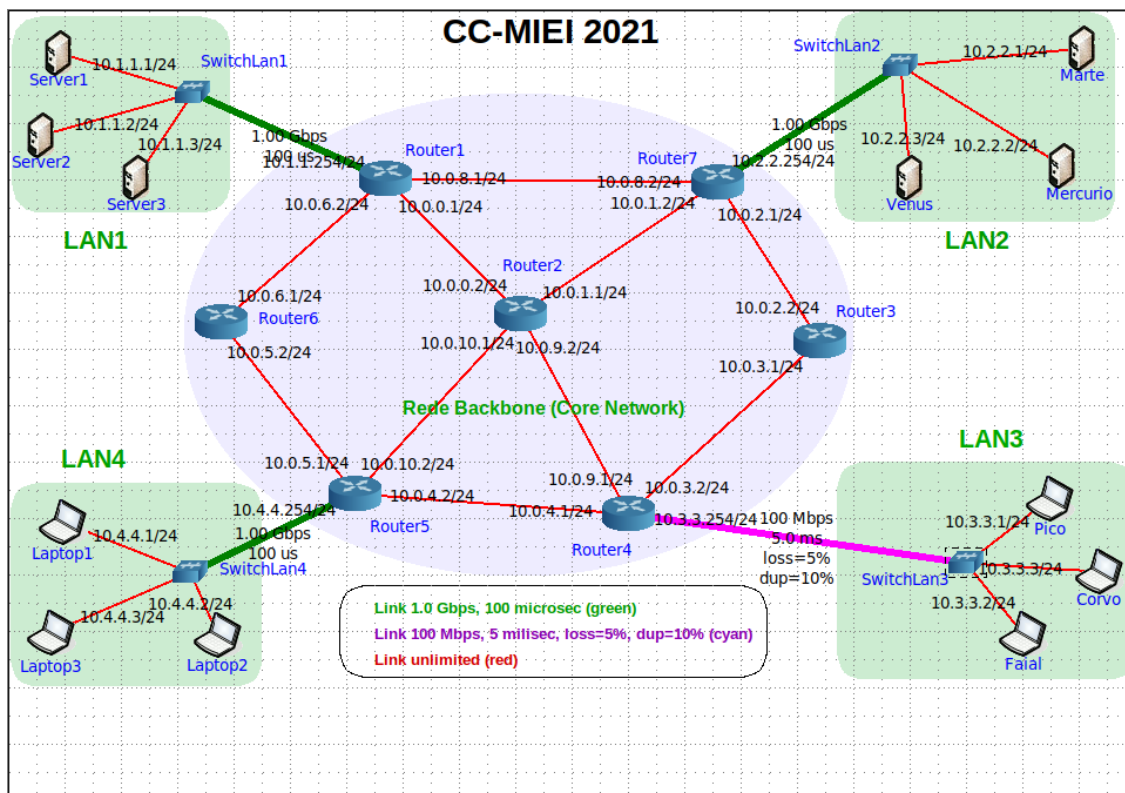


Fig. 15: Implementação da base de dados do servidor de nomes.

Servidor Primário.

Para a implementação da base de dados, tivemos de ter em consideração as informações contidas no enunciado. Com isto, começamos por definir o administrador do domínio `PL04G06.cc.pt`. De seguida, definimos o domínio de nomes com o servidor primário (*Server1*) e o servidor secundário (*Mercurio*). Ainda no domínio de nomes, inserimos também os servidores de email (*Server2*) e (*Server3*) sendo que o primeiro é o principal, visto que tem um menor nível de preferência. A seguir, definimos os *aliases* dos servidores primário e secundário (*ns*) para o servidor primário e *ns2* para o servidor secundário). O *Laptop1* também tem um *alias* definido como *g06*. Posteriormente, registamos os hosts *Marte*, *Venus* e o servidor *Mercurio* com os seus IP's respetivos. Para concluir, definimos que o domínio tem um servidor web e e-mail no *Server2* e que servidor *pop* e *imap* é o *Server3*.

```
core@xubuncore:~/primario$ cat db.cc.pt
$TTL 604800
@      IN      SOA      Server1.cc.pt.  PL04G06.cc.pt. (
                        3          ; Serial
                        604800     ; Refresh
                        86400      ; Retry
                        2419200    ; Expire
                        604800 )   ; Negative Cache TTL
;
@      IN      NS       Server1
@      IN      NS       Mercurio
;
@      IN      MX       10    Server2
@      IN      MX       20    Server3
;
Server1      IN      A       10.1.1.1
ns           IN      CNAME   Server1
;
Mercurio     IN      A       10.2.2.2
ns2          IN      CNAME   Mercurio
;
Laptop1     IN      A       10.4.4.1
g06         IN      CNAME   Laptop1
;
Marte       IN      A       10.2.2.1
Venus       IN      A       10.2.2.3
Mercurio    IN      A       10.2.2.2
;
Server2     IN      A       10.1.1.2
www         IN      CNAME   Server2.cc.pt.
mail        IN      CNAME   Server2.cc.pt.
;
Server3     IN      A       10.1.1.3
pop         IN      CNAME   Server3.cc.pt.
imap        IN      CNAME   Server3.cc.pt.
;
```

Fig. 16: Implementação da base de dados do servidor de nomes.

Para a implementação da base de dados reversa, começamos por inserir no domínio de nomes os servidores primário e secundário. De seguida, utilizamos um *pointer record* (PTR) para registar as diferentes entidades de cada sub-rede. Um *pointer record* fornece o nome de domínio associado a um endereço IP, é exatamente o oposto do registo 'A', que fornece o endereço IP associado a um nome de domínio.

Quando um utilizador tenta aceder a um domínio de nome, ocorre uma pesquisa DNS, correspondendo ao nome de domínio ao endereço IP. Uma consulta reversa de DNS é o oposto desse processo: é uma consulta que começa com o endereço IP e procura o domínio de nome.

Enquanto os registos DNS 'A' são armazenados sob o nome de domínio fornecido, os registos DNS PTR são armazenados no endereço IP - invertido e com ".in-addr.arpa" adicionado. Por exemplo, o registo PTR do endereço IP 10.1.1 seria armazenado em "10.1.1.in-addr.arpa".

Com isto em mente, criamos quatro ficheiros que correspondem às quatro zonas da topologia.

```

core@xubuncore:~/primario$ cat db.1-1-10.rev
$TTL      604800
@         IN      SOA      Server1.cc.pt.  PL04G06.cc.pt. (
                        3      ; Serial
                        604800   ; Refresh
                        86400    ; Retry
                        2419200  ; Expire
                        604800 ) ; Negative Cache TTL
;
@         IN      NS       Server1.cc.pt.
@         IN      NS       Mercurio.cc.pt.

1         IN      PTR      Server1.cc.pt.
2         IN      PTR      Server2.cc.pt.
3         IN      PTR      Server3.cc.pt.

core@xubuncore:~/primario$ cat db.2-2-10.rev
$TTL      604800
@         IN      SOA      Server1.cc.pt.  PL04G06.cc.pt. (
                        3      ; Serial
                        604800   ; Refresh
                        86400    ; Retry
                        2419200  ; Expire
                        604800 ) ; Negative Cache TTL
;
@         IN      NS       Server1.cc.pt.
@         IN      NS       Mercurio.cc.pt.

1         IN      PTR      Marte.cc.pt.
2         IN      PTR      Mercurio.cc.pt.
3         IN      PTR      Venus.cc.pt.

```

Fig. 17: Implementação dos dados do domínio reverse para as LANs 1 e 2.

```

core@xubuncore:~/primario$ cat db.3-3-10.rev
$TTL      604800
@         IN      SOA      Server1.cc.pt.  PL04G06.cc.pt. (
                        3      ; Serial
                        604800   ; Refresh
                        86400    ; Retry
                        2419200  ; Expire
                        604800 ) ; Negative Cache TTL
;
@         IN      NS       Server1.cc.pt.
@         IN      NS       Mercurio.cc.pt.

1         IN      PTR      Pico.cc.pt.
2         IN      PTR      Faial.cc.pt.
3         IN      PTR      Corvo.cc.pt.

core@xubuncore:~/primario$ cat db.4-4-10.rev
$TTL      604800
@         IN      SOA      Server1.cc.pt.  PL04G06.cc.pt. (
                        3      ; Serial
                        604800   ; Refresh
                        86400    ; Retry
                        2419200  ; Expire
                        604800 ) ; Negative Cache TTL
;
@         IN      NS       Server1.cc.pt.
@         IN      NS       Mercurio.cc.pt.

1         IN      PTR      Laptop1.cc.pt.
2         IN      PTR      Laptop2.cc.pt.
3         IN      PTR      Laptop3.cc.pt.

```

Fig. 18: Implementação dos dados do domínio reverse para as LANs 3 e 4.

Foi necessário efetuar algumas alterações ao ficheiro *named.conf* de modo a incluir as diferentes zonas do domínio. As zonas consideradas dizem respeito às diferentes LANs da topologia (de 1 a 4) e à base de dados. Cada zona está etiquetada com *type master* de forma a denotar o servidor primário e permite as transferências por parte do servidor secundário.

```

core@xubuncore:~/primario$ cat 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 "/home/core/primario/named.conf.options";
include "/home/core/primario/named.conf.local";
include "/home/core/primario/named.conf.default-zones";

zone "cc.pt" {
    type master;
    file "/home/core/primario/db.cc.pt";
    allow-transfer{10.2.2.2;};
};

zone "1.1.10.in-addr.arpa." {
    type master;
    file "/home/core/primario/db.1-1-10.rev";
    allow-transfer{10.2.2.2;};
};

zone "2.2.10.in-addr.arpa." {
    type master;
    file "/home/core/primario/db.2-2-10.rev";
    allow-transfer{10.2.2.2;};
};

zone "3.3.10.in-addr.arpa." {
    type master;
    file "/home/core/primario/db.3-3-10.rev";
    allow-transfer{10.2.2.2;};
};

zone "4.4.10.in-addr.arpa." {
    type master;
    file "/home/core/primario/db.4-4-10.rev";
    allow-transfer{10.2.2.2;};
};

```

Fig. 19: Conteúdo do ficheiro named.conf.

O ficheiro *named.conf.options* também requiriu algumas alterações na medida em que foi necessário acrescentar novos servidores como *forwarders* de forma a encaminhar *queries* DNS para o exterior.

```

core@xubuncore:~/primario$ cat 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;
        193.136.9.240;
        193.136.19.1;
    };

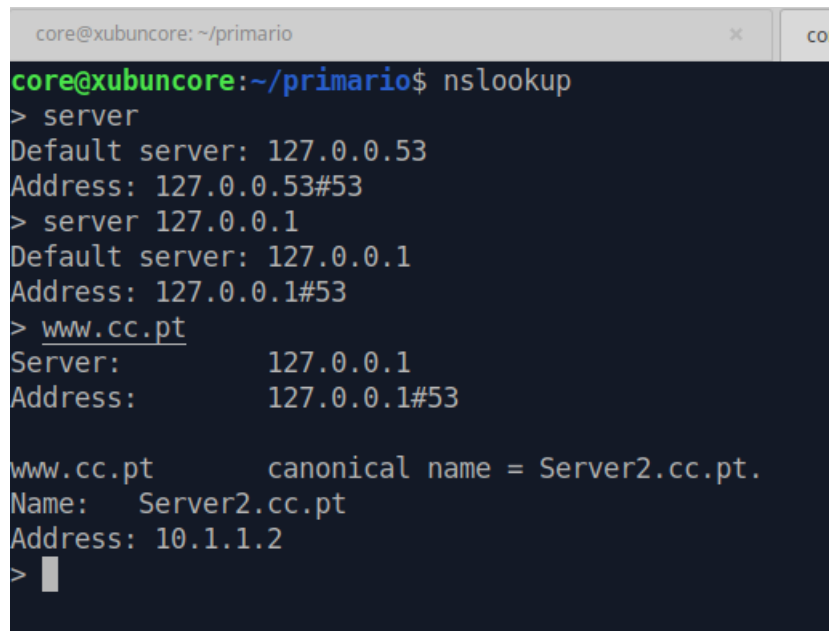
    //=====
    // If BIND logs error messages about the root key being expired,
    // you will need to update your keys.  See https://www.isc.org/bind-keys
    //=====
    dnssec-validation auto;

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

```

Fig. 20: Conteúdo do ficheiro named.conf.options.

A seguinte figura retrata um dos testes efetuados que trata de interrogar o *localhost* acerca do endereço *www.cc.pt*. Tal como podemos observar na seguinte figura, a resposta obtida está de acordo com o implementado.

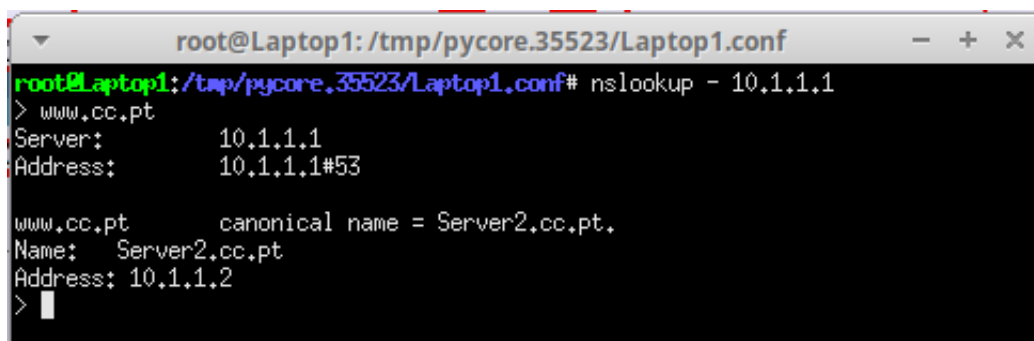
A terminal window titled 'core@xubuncore: ~/primario' with a dark background. The user enters 'nslookup' at the prompt. The output shows the default server as 127.0.0.53. Then, the user enters 'server 127.0.0.1', and the output shows the default server as 127.0.0.1. Finally, the user enters 'www.cc.pt', and the output shows the server as 127.0.0.1 and the address as 127.0.0.1#53. Below this, it shows the canonical name as Server2.cc.pt, the name as Server2.cc.pt, and the address as 10.1.1.2.

```
core@xubuncore: ~/primario$ nslookup
> server
Default server: 127.0.0.53
Address: 127.0.0.53#53
> server 127.0.0.1
Default server: 127.0.0.1
Address: 127.0.0.1#53
> www.cc.pt
Server:          127.0.0.1
Address:         127.0.0.1#53

www.cc.pt        canonical name = Server2.cc.pt.
Name:   Server2.cc.pt
Address: 10.1.1.2
>
```

Fig. 21: Teste do servidor primário fora do emulador CORE.

Além disso, também foram efetuados testes na topologia core, efetuando uma *query* ao servidor primário a partir de um *host* de cada LAN, utilizando o *nslookup*.

A terminal window titled 'root@Laptop1: /tmp/pycore.35523/Laptop1.conf' with a dark background. The user enters 'nslookup - 10.1.1.1'. The output shows the server as 10.1.1.1 and the address as 10.1.1.1#53. Below this, it shows the canonical name as Server2.cc.pt, the name as Server2.cc.pt, and the address as 10.1.1.2.

```
root@Laptop1: /tmp/pycore.35523/Laptop1.conf# nslookup - 10.1.1.1
> www.cc.pt
Server:          10.1.1.1
Address:         10.1.1.1#53

www.cc.pt        canonical name = Server2.cc.pt.
Name:   Server2.cc.pt
Address: 10.1.1.2
>
```

Fig. 22: Teste no Laptop1 para o servidor primário.

```
> set q=mx
> cc.pt
Server:      10.1.1.1
Address:     10.1.1.1#53

cc.pt  mail exchanger = 10 Server2.cc.pt.
cc.pt  mail exchanger = 20 Server3.cc.pt.
>
```

Fig. 23: Teste do *nslookup* aos emails.

```
core@xubuncore:~$ nslookup www.cc.pt
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
Name:   www.cc.pt
Address: 185.53.177.10
```

Fig. 24: Teste do *nslookup* antes de alterar as configurações do ficheiro */etc/resolv.conf*.

```
core@xubuncore:~$ sudo vim /etc/resolv.conf
core@xubuncore:~$ nslookup www.cc.pt
;; connection timed out; no servers could be reached
```

Fig. 25: Teste do *nslookup* após alterar as configurações do ficheiro */etc/resolv.conf*.

Servidor Secundário.

Para a implementação do servidor secundário foi necessário editar o ficheiro de configuração do DNS, *named.conf*, e também o ficheiro *named.conf.options* que contém todas as opções de configuração. Assim, o ficheiro *named.conf* inclui diferentes zonas desde a base de dados até às diferentes LANs (1 a 4) da topologia de rede e fornece permissões de transferência ao servidor primário. Além disso, as zonas estão etiquetadas com *slave* de forma a denotar que se trata de um servidor *backup*.

Para o ficheiro *named.conf.options* foi necessário adicionar 2 *forwarders* de forma a permitir encaminhar as *queries* DNS para servidores externos.

Nas figuras 26 e 27 é possível observar o conteúdo destes ficheiros.

```
core@xubuncore:~/secundario$ cat 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 "/home/core/secundario/named.conf.options";
include "/home/core/secundario/named.conf.local";
include "/home/core/secundario/named.conf.default-zones";

zone "cc.pt" {
    type slave;
    file "/var/cache/bind/db.cc.pt";
    masters { 10.1.1.1; };
};

zone "1.1.10.in-addr.arpa" {
    type slave;
    file "/var/cache/bind/db.1-1-10.rev";
    masters { 10.1.1.1; };
};

zone "2.2.10.in-addr.arpa" {
    type slave;
    file "/var/cache/bind/db.2-2-10.rev";
    masters { 10.1.1.1; };
};

zone "3.3.10.in-addr.arpa" {
    type slave;
    file "/var/cache/bind/db.3-3-10.rev";
    masters { 10.1.1.1; };
};

zone "4.4.10.in-addr.arpa" {
    type slave;
    file "/var/cache/bind/db.4-4-10.rev";
    masters { 10.1.1.1; };
};
```

Fig. 26: Conteúdo do ficheiro *named.conf* do servidor secundario.

```
core@xubuncore:~/secundario$ cat 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 {
        193.136.9.240;
        193.136.19.1;
    };

    //=====  

    // 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; };
};
```

Fig. 27: Conteúdo do ficheiro *named.conf.options* do servidor secundario.

De modo a verificar se o programa funciona corretamente, foram efetuados alguns testes. Na topologia de rede começou-se por arrancar os servidores primário e secundário. De seguida, efetuou-se um teste utilizando o *nslookup* para o endereço IP do servidor secundário a partir de um *host* de cada LAN. É possível observar os resultados obtidos a partir de cada *host* nas seguintes imagens.

```
root@Laptop1:/tmp/pycore.35523/Laptop1.conf# nslookup - 10.2.2.2
> www.cc.pt
Server:      10.2.2.2
Address:     10.2.2.2#53

www.cc.pt    canonical name = Server2.cc.pt.
Name:   Server2.cc.pt
Address: 10.1.1.2
```

Fig. 28: Teste do *nslookup* no Laptop1 com o IP 10.2.2.2

```
root@Pico:/tmp/pycore.35523/Pico.conf# nslookup - 10.2.2.2
> www.cc.pt
Server:      10.2.2.2
Address:     10.2.2.2#53

www.cc.pt    canonical name = Server2.cc.pt.
Name:   Server2.cc.pt
Address: 10.1.1.2
>
```

Fig. 29: Teste do *nslookup* no Pico com o IP 10.2.2.2

```
root@Server2:/tmp/pycore.35523/Server2.conf# nslookup - 10.2.2.2
> www.cc.pt
Server:      10.2.2.2
Address:     10.2.2.2#53

www.cc.pt    canonical name = Server2.cc.pt.
Name:   Server2.cc.pt
Address: 10.1.1.2
```

Fig. 30: Teste do *nslookup* no Server com o IP 10.2.2.2

```
root@Venus:/tmp/pycore.35523/Venus.conf# nslookup - 10.2.2.2
> www.cc.pt
Server:      10.2.2.2
Address:     10.2.2.2#53

www.cc.pt    canonical name = Server2.cc.pt.
Name:   Server2.cc.pt
Address: 10.1.1.2
```

Fig. 31: Teste do *nslookup* no Venus com o IP 10.2.2.2

Conclusão

Com a resolução do presente trabalho prático conseguimos consolidar os conceitos leccionados nas aulas teóricas e, por conseguinte, aprofundar conhecimentos relacionados com DNS (Serviço de Resolução de Nomes).

Numa primeira etapa, denominada como Questões e Respostas, o trabalho realizado focou-se em consultas ao serviço de nomes DNS, neste sentido foram usados clientes de DNS como o `host`, o `dig` e o `nslookup`. Em particular, as soluções apresentadas usam sobretudo o `nslookup` e são maioritariamente resolvidas através de queries simples, que utilizam registos como : `A` para descobrir endereços IPv4, `AAAA` para saber endereços IPv6, `MX` para identificar os servidores de mail para um domínio, `NS` como forma de desvendar os servidores que detem autoridade numa zona, `SOA` para obter o SOA record e todas as suas pertinentes informações, bem como o `ANY` para todas as informações. Em adição, ficamos com algumas noções pertinentes como a distinção entre host e domínio, respostas autoritativas e não autoritativas e a par de especificidades como o facto de um endereço de email principal tem um número menor para destacar o seu maior grau de prioridade ou até mesmo o facto de que nos secundários existem mecanismos que permitem a atualização automática através do primário (mecanismo de transferência de zona).

Finalmente, na segunda etapa, passamos à instalação, configuração e teste de um domínio CC.PT. Inicialmente, apresentamos a topologia da rede a qual serviu de base para a elaboração do domínio em questao. Assim sendo, partimos pela implementação do servidor primário, começando pelas bases de dados (normal e inversas), seguidamente atualizamos os ficheiros `named.conf` e `named.conf.options`, e por fim terminamos com alguns testes e exemplos demonstrativos das funcionalidades e requisitos pretendidos. Analogamente acontece com o servidor secundário, começamos pela modificação dos ficheiros `named.conf` e `named.conf.options`, e por fim realizamos testes de forma a comprovar o correto funcionamento da implementação.