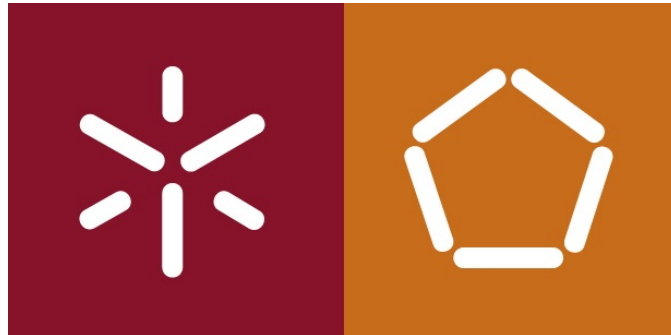


UNIVERSIDADE DO MINHO

MESTRADO INTEGRADO EM ENGENHARIA INFORMÁTICA



Comunicações por Computadores

RELATÓRIO DO TRABALHO PRÁTICO 3

PROTOCOLOS DA CAMADA DE TRANSPORTE

PL 6 GRUPO 3



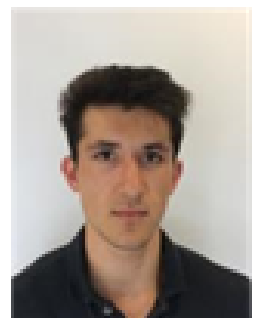
Pedro Freitas

A80975



Nuno Silva

A78156



Shahzod Yusupov

A82617

April 5, 2019

Parte I - Consultas ao serviço de nomes DNS

Com base no manual das aplicações (ex: `man nslookup` ou `man dig`) e no material de suporte procure responder às seguintes questões (incompletas para já, servem apenas de exemplos genéricos a que deve saber responder):

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

Este ficheiro contém a informação que determina os parâmetros operacionais dos servidores DNS.

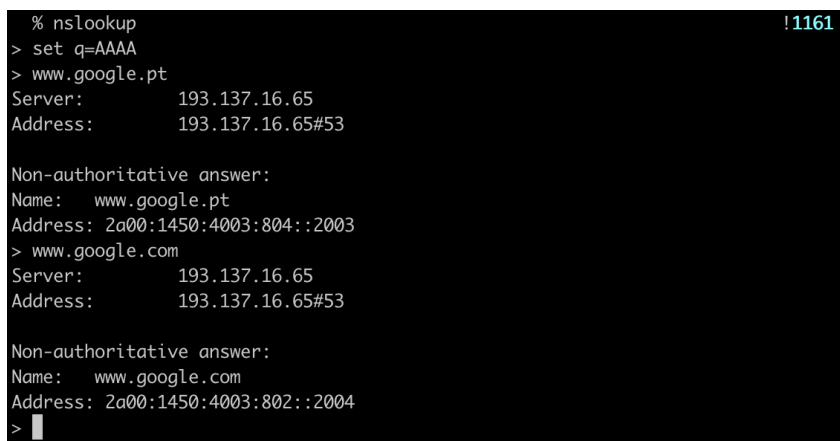
Estes servidores permitem que as aplicações que correm no sistema operativo traduzam nomes de domínios para endereços IP, que serão necessários para aceder a recursos de uma área local ou da Internet.

b) Os servidores *www.google.pt* e *www.google.com* têm endereços IPv6? Se sim, quais?

Sim. Como vemos pela figura 1 tanto o primeiro como o segundo endereço têm endereços IPv6.

www.google.pt - 2a00:1450:4003:804::2003

www.google.com - 2a00:1450:4003:802::2004



```
% nslookup !1161
> set q=AAAA
> www.google.pt
Server:      193.137.16.65
Address:     193.137.16.65#53

Non-authoritative answer:
Name:   www.google.pt
Address: 2a00:1450:4003:804::2003
> www.google.com
Server:      193.137.16.65
Address:     193.137.16.65#53

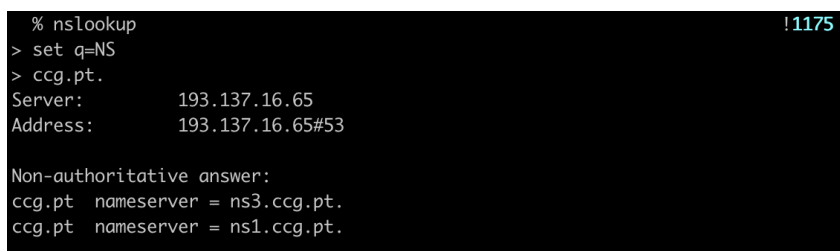
Non-authoritative answer:
Name:   www.google.com
Address: 2a00:1450:4003:802::2004
>
```

Figure 1

c) Quais os servidores de nomes definidos para os seguintes domínios: “cpg.pt.”, “pt.” e “.”?

Todos os servidores de nomes definidos para o domínio *cpg.pt.*

- ns3.cpg.pt.
- ns1.cpg.pt.



```
% nslookup !1175
> set q=NS
> cpg.pt.
Server:      193.137.16.65
Address:     193.137.16.65#53

Non-authoritative answer:
cpg.pt nameserver = ns3.cpg.pt.
cpg.pt nameserver = ns1.cpg.pt.
```

Figure 2

Todos os servidores de nomes definidos para o domínio *pt.*

- g.dns.pt.

- f.dns.pt.
- e.dns.pt.
- a.dns.pt.
- ns.dns.br.
- d.dns.pt.
- ns2.nic.fr.
- sns-pb.isc.org.
- c.dns.pt.
- b.dns.pt.

```
% nslookup !1181
> set q=NS
> pt.
Server:      193.137.16.65
Address:     193.137.16.65#53

Non-authoritative answer:
pt      nameserver = g.dns.pt.
pt      nameserver = f.dns.pt.
pt      nameserver = e.dns.pt.
pt      nameserver = a.dns.pt.
pt      nameserver = ns.dns.br.
pt      nameserver = d.dns.pt.
pt      nameserver = ns2.nic.fr.
pt      nameserver = sns-pb.isc.org.
pt      nameserver = c.dns.pt.
pt      nameserver = b.dns.pt.
```

Figure 3

Todos os servidores de nomes definidos para o domínio .

- a.root-servers.net
- m.root-servers.net
- k.root-servers.net
- c.root-servers.net
- b.root-servers.net
- g.root-servers.net
- e.root-servers.net
- f.root-servers.net
- l.root-servers.net
- i.root-servers.net
- j.root-servers.net
- d.root-servers.net
- h.root-servers.net

```
% nslookup !1179
> set q=NS
> .
Server:      193.137.16.65
Address:     193.137.16.65#53

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

Figure 4

d) Existe o domínio *eureka.software*. ? Será que *eureka.software*. é um host ?

Como vemos pela Figura 6 , *eureka.software*. tem name servers logo podemos concluir que é um domínio.

Já pela Figura 7 vemos que este tem endereço IP (34.214.90.141) logo também é um host.

```
> eureka.software.  
Server:      193.137.16.65  
Address:     193.137.16.65#53  
  
Non-authoritative answer:  
eureka.software nameserver = ns-1241.awsdns-27.org.  
eureka.software nameserver = ns-312.awsdns-39.com.  
eureka.software nameserver = ns-957.awsdns-55.net.  
eureka.software nameserver = ns-1624.awsdns-11.co.uk.  
  
Authoritative answers can be found from:  
ns-312.awsdns-39.com    internet address = 205.251.193.56  
ns-957.awsdns-55.net   internet address = 205.251.195.189  
> █
```

Figure 5

```
% nslookup !1185  
> set q=A  
> eureka.software.  
Server:      193.137.16.65  
Address:     193.137.16.65#53  
  
Non-authoritative answer:  
Name:   eureka.software  
Address: 34.214.90.141  
> █
```

Figure 6

e) Qual é o servidor DNS primário definido para o domínio *ami.pt*. ? Aceita queries recursivas? Porquê?

Como vemos pela figura X o servidor DNS primário do domínio *ami.pt*. é *ns1.dot2web.com* .

```
% nslookup !1189  
> set q=SOA  
> ami.pt  
Server:      193.137.16.65  
Address:     193.137.16.65#53  
  
Non-authoritative answer:  
ami.pt  
    origin = ns1.dot2web.com  
    mail addr = dc.dot2web.pt  
    serial = 2019021301  
    refresh = 3600  
    retry = 7200  
    expire = 1209600  
    minimum = 86400
```

Figure 7

Depois de questionarmos o domínio, temos de tentar questionar o servidor DNS primário do mesmo.
Como obtivemos resposta, este aceita queries recursivas:

```
% dig ns1.dot2web.com

;; <<>> DiG 9.14.0 <<>> ns1.dot2web.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 52979
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 2048
;; QUESTION SECTION:
;ns1.dot2web.com.                IN      A

;; ANSWER SECTION:
ns1.dot2web.com.                1277    IN      A      80.172.230.28
```

Figure 8

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

```
pedrofreitas@MacBook-Pro-de-Pedro ~
% dig +short SOA ami.pt. | cut -d ' ' -f1
ns1.dot2web.com.
```

Figure 9

g) Onde são entregues as mensagens dirigidas a *marcelo@presidencia.pt* ? E a *guterres@onu.org* ?

As mensagens dirigidas a *marcelo@preseidencia.pt* são entregues ou em *mail1.presidencia.pt* ou em *mail2.presidencia.pt* .

As mensagens dirigidas a *guterres@onu.org* são entregues em *mail.onu.org* .

```
% nslookup !1179
> set q=NS
> .
Server:      193.137.16.65
Address:     193.137.16.65#53

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

Figure 10

h) Que informação é possível obter acerca de *www.whitehouse.gov*? Qual é o endereço IPv4 associado?

É possível obter o mail adress: *hostmaster.akamai.com* .

Como podemos pela figura 11 o endereço IPv4 associado é: 193.137.16.65

```
% nslookup !2057
> set q=SOA
> www.whitehouse.gov
Server:      193.137.16.65
Address:     193.137.16.65#53

Non-authoritative answer:
www.whitehouse.gov      canonical name = wildcard.whitehouse.gov.edgekey.net.
wildcard.whitehouse.gov.edgekey.net canonical name = e4036.dscb.akamaiedge.net.

Authoritative answers can be found from:
dscb.akamaiedge.net
    origin = n0dscb.akamaiedge.net
    mail addr = hostmaster.akamai.com
    serial = 1553587771
    refresh = 1000
    retry = 1000
    expire = 1000
    minimum = 1800
```

Figure 11

```
% nslookup
> set q=A
> www.whitehouse.gov
Server:          193.137.16.65
Address:         193.137.16.65#53
```

Figure 12

i) Consegue interrogar o DNS sobre o endereço IPv6 2001:690:a00:1036:1113::247 usando algum dos clientes DNS? Que informação consegue obter? Supondo que teve problemas com esse endereço, consegue obter um contacto do responsável por esse IPv6?

```
% nslookup 2001:690:a00:1036:1113::247 !2086
7.4.2.0.0.0.0.0.0.0.0.0.3.1.1.1.6.3.0.1.0.0.a.0.0.9.6.0.1.0.0.2.ip6.arpa n
ame = www.fccn.pt.

Authoritative answers can be found from:
0.9.6.0.1.0.0.2.ip6.arpa      nameserver = ns02.fccn.pt.
0.9.6.0.1.0.0.2.ip6.arpa      nameserver = ns03.fccn.pt.
0.9.6.0.1.0.0.2.ip6.arpa      nameserver = ns01.fccn.pt.
ns02.fccn.pt      internet address = 193.136.2.228
ns02.fccn.pt      has AAAA address 2001:690:a80:4001::200
```

Figure 13

Como vemos pela Figura 12 podemos obter alguns servidores de nomes e o servidor *www.fccn.pt* . Com esse nome podemos procurar o domínio dele sendo ele *fccn.pt* .

Assim podemos concluir que o contacto responsável pelo IPv6 é o próprio domínio *fccn.pt* , como podemos ver pela Figura 13.

```
% nslookup !2091
> set q=AAAA
> fccn.pt
Server:          193.137.16.65
Address:         193.137.16.65#53

Non-authoritative answer:
Name:   fccn.pt
Address: 2001:690:a00:1036:1113::247
```

Figure 14

j) Os secundários usam um mecanismo designado por “Transferência de zona” para se atualizarem automaticamente a partir do primário, usando os parâmetros definidos no Record do tipo SOA do domínio. Descreve sucintamente esse mecanismo com base num exemplo concreto (ex: *di.uminho.pt* ou o domínio *cc.pt* que vai ser criado na topologia virtual).

A transferência de zona é um tipo de transação disponível de DNS. Esta transferência é uma espécie de transação entre um cliente e um servidor, usando TCP, e é muito usado por admins para replicar as suas bases de dados. Assim os parâmetros definidos no Record do tipo SOA do domínio tornam-se essenciais e é por isso necessário entender cada campo.

Serial - Representa o número de série da zona. Se o servidor secundário que lhe está associado notar uma

diminuição deste valor ele entende que a zona está desatualizada e por isso executa uma transferência de zona.

Refresh - Depois de detetar alterações da zona o servidor secundário deve esperar estes segundos para contactar o primário e atualizar

Retry - Número de segundos para uma segunda tentativa de contacto com o primário, caso uma tentativa tenha falhado. Devido ao parâmetro refresh representar o tempo de espera também, este tem de ser menor.

Expire - Número de segundos após o qual o servidor secundário deve para de solicitar o primário se não obtiver resposta. Este valor então tem de ser superior aos dois parâmetros anteriores.

Assim temos como exemplo os parâmetros do domínio cc.pt. implementado na topologia virtual:

```
@      IN      SOA      dns.cc.pt. grupo63.cc.pt. (
                                1          ; Serial
                                604800     ; Refresh
                                86400      ; Retry
                                2419200    ; Expire
                                604800 )   ; Negative Cache TTL
.
```

Figure 15

Parte II - Instalação, configuração e teste de um domínio CC.PT

Objetivos

Pretende-se que crie um domínio CC.PT para a topologia de rede que estamos a usar nas aulas práticas (CC-Topo- 2019.imn), de modo a que se possam usar os nomes em vez dos endereços IP. No final deve, por exemplo, poder fazer-se “ping servidor1.cc.pt” ou mesmo apenas “ping Servidor1” ou “ping Servidor1.cc.pt.”em vez de “ping 10.1.1.1”.

Requisitos

- Criação do domínio cc.pt co servidor primário em Servidor1 10.1.1.1 e secundário em Urano 10.2.2.3
- Criação do domínio reverso 1.1.10.in-addr.arpa com os mesmos servidores
- o servidor primário do domínio é o “Servidor1” com endereço 10.1.1.1, também designado por dns.cc.pt, tendo como secundário o “Urano” com endereço 10.2.2.3, com alias dns2.cc.pt. O administrador do domínio é o grupoXX@cc.pt (onde XX é o número do grupo).
- O domínio tem também um servidor Web (www.cc.pt) e um servidor de e-mail principal (mail.cc.pt) em Servidor3. O servidor pop e imap é o Servidor2, que é também servidor secundário do e-mail para o domínio;
- Sem prejuízo de outros registos que se possam considerar, devem estar registados também o Cliente1.cc.pt com alias GrupoXX.cc.pt onde XX é o número do grupo, e Alfa.cc.pt, Beta.cc.pt e Gama.cc.pt no domínio de nomes e no domínio reverso.

Pré requisitos

Antes de começar certifique-se que o software BIND9 está instalado (*sudo apt-get install bind9*). Este pacote vem já pré-configurado de base (ficheiros em */etc/bind*) pelo que o número de alterações a efetuar é mínimo.

Preparativos especiais para ambiente CORE

Passo 1) replicar ficheiros de configuração

Para não criar conflitos, e uma vez que todos os nós da topologia CORE usam na realidade o mesmo filesystem, o primeiro passo é copiar os ficheiros de configuração para uma nova pasta. Sugere-se que use por exemplo **\$HOME/primario**, para o primário e **\$HOME/secundario** para o servidor secundário, onde \$HOME é a diretoria *default* do utilizador (no caso do user *core*, será */home/core*):

```
Terminal - core@XubunCORE: ~
File Edit View Terminal Go Help
/home/core
core@XubunCORE:~$ rsync -av /etc/bind/ ~/primario/
sending incremental file list
created directory /home/core/primario
./
bind.keys
db.0
db.127
db.255
db.empty
db.local
db.root
named.conf
named.conf.default-zones
named.conf.local
named.conf.options
rsync: send_files failed to open "/etc/bind/rndc.key": Permission denied (13)
zones.rfc1918

sent 10869 bytes  received 262 bytes  22262.00 bytes/sec
total size is 10153  speedup is 0.91
rsync error: some files/attrs were not transferred (see previous errors) (code 2
3) at main.c(1070) [sender=3.0.9]
core@XubunCORE:~$
```

Figure 16: Cópia dos ficheiros para a pasta **primario**

```
Terminal - core@XubunCORE: ~
File Edit View Terminal Go Help
core@XubunCORE:~$ rsync -av /etc/bind/ ~/secundario/
sending incremental file list
created directory /home/core/secundario
./
bind.keys
db.0
db.127
db.255
db.empty
db.local
db.root
named.conf
named.conf.default-zones
named.conf.local
named.conf.options
rsync: send_files failed to open "/etc/bind/rndc.key": Permission denied (13)
zones.rfc1918

sent 10869 bytes  received 262 bytes  22262.00 bytes/sec
total size is 10153  speedup is 0.91
rsync error: some files/attrs were not transferred (see previous errors) (code 2
3) at main.c(1070) [sender=3.0.9]
core@XubunCORE:~$
```

Figure 17: Cópia dos ficheiros para a pasta **secundario**

Podemos ver que os ficheiros foram copiados para as duas pastas nas duas seguintes imagens:

```
Terminal - core@XubunCORE: ~/primario
File Edit View Terminal Go Help
core@XubunCORE:~$ cd primario/
core@XubunCORE:~/primario$ ls
bind.keys  db.255    db.root      named.conf.local
db.0       db.empty  named.conf    named.conf.options
db.127     db.local  named.conf.default-zones  zones.rfc1918
core@XubunCORE:~/primario$
```

Figure 18: *ls* primario

```
Terminal - core@XubunCORE: ~/secundario
File Edit View Terminal Go Help
core@XubunCORE:~$ cd secundario/
core@XubunCORE:~/secundario$ ls
bind.keys  db.255    db.root      named.conf.local
db.0       db.empty  named.conf    named.conf.options
db.127     db.local  named.conf.default-zones  zones.rfc1918
core@XubunCORE:~/secundario$
```

Figure 19: *ls* secundario

Passo 2) parar o servidor DNS pré-instalado

De forma a podermos parar o servidor DNS pré-instalado executamos os comandos presentes na seguinte imagem:

```
Terminal - core@XubunCORE: ~
File Edit View Terminal Go Help
core@XubunCORE:~$ sudo /etc/init.d/bind9 status
* bind9 is running
core@XubunCORE:~$ sudo /etc/init.d/bind9 stop
* Stopping domain name service... bind9
waiting for pid 1245 to die

core@XubunCORE:~$ sudo /etc/init.d/bind9 status
* bind9 is not running
core@XubunCORE:~$
```

Figure 20: Comandos executados para parar o servidor dns pré-instalado

Passo 3) reconfigurar apparmor para permitir que /usr/sbin/named aceda a ficheiros noutros locais

O kernel Linux inclui um sistema de proteção para evitar que alguns programas acessem a ficheiros que não devem! Para isso deve-se verificar se o daemon respetivo (**named**) consta na lista de perfis controlados pelo apparmor:

```
Terminal - core@XubunCORE: ~
File Edit View Terminal Go Help
core@XubunCORE:~$ sudo /etc/init.d/apparmor status
apparmor module is loaded.
17 profiles are loaded.
17 profiles are in enforce mode.
  /sbin/dhclient
  /usr/bin/evince
  /usr/bin/evince-previewer
  /usr/bin/evince-previewer//launchpad_integration
  /usr/bin/evince-previewer//sanitized_helper
  /usr/bin/evince-thumbnailer
  /usr/bin/evince-thumbnailer//sanitized_helper
  /usr/bin/evince//launchpad_integration
  /usr/bin/evince//sanitized_helper
  /usr/lib/NetworkManager/nm-dhcp-client.action
  /usr/lib/connman/scripts/dhclient-script
  /usr/lib/cups/backend/cups-pdf
  /usr/lib/lightdm/lightdm/lightdm-guest-session-wrapper
  /usr/lib/lightdm/lightdm/lightdm-guest-session-wrapper//chromium_browser
  /usr/sbin/cupsd
  /usr/sbin/named
  /usr/sbin/tcpdump
0 profiles are in complain mode.
2 processes have profiles defined.
2 processes are in enforce mode.
  /sbin/dhclient (984)
  /usr/sbin/cupsd (701)
0 processes are in complain mode.
0 processes are unconfined but have a profile defined.
core@XubunCORE:~$
```

Figure 21: Lista de perfis controlados pelo apparmor

Neste caso vamos ter de reconfigurar essas permissões para que o `/usr/sbin/named` possa ler as novas directorias.

- 1) `/etc/apparmor.d/usr.sbin.named` com vista a acrescentar duas novas linhas de permissões) :

```
core@XubunCORE:~$ cat /etc/apparmor.d/usr.sbin.named
# vim:syntax=apparmor
# Last Modified: Fri Jun 1 16:43:22 2007
#include <tunables/global>

/usr/sbin/named {
  #include <abstractions/base>
  #include <abstractions/nameservice>

  capability net_bind_service,
  capability setgid,
  capability setuid,
  capability sys_chroot,
  capability sys_resource,

  # /etc/bind should be read-only for bind
  # /var/lib/bind is for dynamically updated zone (and journal) files.
  # /var/cache/bind is for slave/stub data, since we're not the origin of it.
  # See /usr/share/doc/bind9/README.Debian.gz
  /etc/bind/** r,
  /home/core/primario/** r,
  /home/core/secundario/** r,
  /var/lib/bind/** rw,
  /var/lib/bind/ rw,
  /var/cache/bind/** rw,
  /var/cache/bind/ rw,
```

Figure 22: `ls` primario

- 2) Parar e reiniciar o apparmor:

```
Terminal - core@XubunCORE: ~
File Edit View Terminal Go Help
core@XubunCORE:~$ sudo /etc/init.d/apparmor stop
* Clearing AppArmor profiles cache
All profile caches have been cleared, but no profiles have been unloaded.
Unloading profiles will leave already running processes permanently
unconfined, which can lead to unexpected situations.

To set a process to complain mode, use the command line tool
'aa-complain'. To really tear down all profiles, run the init script
with the 'teardown' option."
core@XubunCORE:~$ sudo /etc/init.d/apparmor start
* Starting AppArmor profiles
Skipping profile in /etc/apparmor.d/disable: usr.bin.firefox
Skipping profile in /etc/apparmor.d/disable: usr.sbin.rsyslogd
```

Figure 23: Para e iniciar apparmor

```
Terminal - core@XubunCORE: ~
File Edit View Terminal Go Help
core@XubunCORE:~$ sudo /etc/init.d/apparmor status
apparmor module is loaded.
17 profiles are loaded.
17 profiles are in enforce mode.
  /sbin/dhclient
  /usr/bin/evince
  /usr/bin/evince-previewer
  /usr/bin/evince-previewer//launchpad_integration
  /usr/bin/evince-previewer//sanitized_helper
  /usr/bin/evince-thumbnailer
  /usr/bin/evince-thumbnailer//sanitized_helper
  /usr/bin/evince//launchpad_integration
  /usr/bin/evince//sanitized_helper
  /usr/lib/NetworkManager/nm-dhcp-client.action
  /usr/lib/connman/scripts/dhclient-script
  /usr/lib/cups/backend/cups-pdf
  /usr/lib/lightdm/lightdm/lightdm-guest-session-wrapper
  /usr/lib/lightdm/lightdm/lightdm-guest-session-wrapper//chromium_browser
  /usr/sbin/cupsd
  /usr/sbin/named
  /usr/sbin/tcpdump
0 profiles are in complain mode.
2 processes have profiles defined.
2 processes are in enforce mode.
  /sbin/dhclient (984)
  /usr/sbin/cupsd (701)
0 processes are in complain mode.
0 processes are unconfined but have a profile defined.
core@XubunCORE:~$
```

Figure 24: status do apparmor

Configuração de Servidores

As configurações a fazer devem respeitar as seguintes regras:

- os dados do domínio **cc.pt** devem ser editados/mantidos no ficheiro **db.cc.pt**
- os dados do domínio reverso **1.1.10.in-addr.arpa.** relativos à rede **10.1.1.0/24** devem ser editados/mantidos no ficheiro **db.1-1-10.rev** (aplicar sempre o mesmo critério de nomes a outros domínios reversos que decida incluir)
- o **servidor primário** do domínio é o “**Servidor1**” com endereço **10.1.1.1**, também designado por **dns.cc.pt**, tendo como secundário o “**Urano**” com endereço **10.2.2.3**, com alias **dns2.cc.pt**. O administrador do domínio é o **grupoXX@cc.pt** (onde **XX** é o número do grupo).
- O domínio tem também um **servidor Web** (**www.cc.pt**) e um servidor de e-mail principal (**mail.cc.pt**) em **Servidor3**. O servidor **pop** e **imap** é o **Servidor2**, que é também **servidor secundário do e-mail** para o domínio;
- Sem prejuízo de outros registos que se possam considerar, devem estar registados também o **Cliente1.cc.pt** com alias **GrupoXX.cc.pt** onde **XX** é o número do grupo, e **Alfa.cc.pt**, **Beta.cc.pt** e **Gama.cc.pt** no domínio de nomes e no domínio reverso.

Configuração do servidor primário

1) Editar o ficheiro `/etc/hosts` para incluir os registos 10.1.1.1 Servidor1 dns.cc.pt do primário e 10.2.2.3 Urano dns2.cc.pt do secundário; este passo é obrigatório para que os servidores DNS se identifiquem corretamente a si próprios:

```
### begin CORE auto-generated hosts entries
10.0.0.1      A0
10.0.0.2      A1
10.0.0.3      A2
10.0.0.4      A3
10.0.0.5      A4
10.0.0.6      A5
10.0.0.7      A6
10.0.0.8      A7
10.0.0.9      A8
10.0.0.10     A9
10.0.0.11     A10
10.0.0.12     A11
10.0.0.13     A12
10.0.0.14     A13
10.0.0.15     A14
10.0.0.16     A15
10.0.0.17     A16
10.1.1.1      Servidor1      dns.cc.pt
10.2.2.3      Urano          dns2.cc.pt
### end CORE auto-generated hosts entries
```

Figure 25: Ficheiro `/etc/hosts`

2) Editar o ficheiro `primario/named.conf.options` por forma a incluir os servidores 193.136.9.240 e 193.136.19.1 (servidores do DI) como forwarders:

```
forwarders {
    193.136.9.240;
    193.136.19.1;
};
```

Figure 26: Ficheiro `primario/named.conf.options`

3) Editar o ficheiro `primario/named.conf` para incluir a indicação das novas zonas “cc.pt”, “1.1.10.in-addr.arpa” etc (ver exemplos em `named.conf.default-zones`) e corrigir os nomes das diretorias, substituindo `/etc/bind/` pela nova diretoria (`/home/core/primario` ou outra equivalente). Não esquecer de incluir uma cláusula “allow-transfer” a dar permissão de transferência da base de dados ao servidor secundário.

Para podermos executar este passo tivemos de ponderar diferentes abordagens a fazer ao enunciado e ao pedido. Depois de analisarmos também a topologia CORE *CC-TOPO-2019.imn* reparamos na existência de 4 LAN’s diferentes. Isto levou-nos a optar por não criar apenas duas zonas, uma para *cc.pt* e outra para a conter todas as outras máquinas ligadas na topologia, mas sim criar 5 zonas: Uma para cada LAN e ainda a dita *cc.pt*. Além disso também sabemos que estamos a tratar das zonas do servidor DNS principal, sabemos que o tipo delas é **master**. Cada zona tem um ficheiro correspondente à sua base de dados e ainda uma cláusula *allow-transfer* que contém o endereço do servidor secundário, para

permitir transferências para o mesmo. Tendo isto em mente o ficheiro *named.conf* contém as seguintes zonas:

```
zone "cc.pt" {  
    type master;  
    file "/home/core/primario/db.cc.pt";  
    allow-transfer {10.2.2.3};  
};
```

Figure 27: *primario/named.conf.options* zona: **cc.pt**

```
zone "1.1.10.in-addr.arpa" {  
    type master;  
    file "/home/core/primario/db.1-1-10.rev";  
    allow-transfer {10.2.2.3};  
};
```

Figure 28: *primario/named.conf.options* zona: **1.1.10.in-addr.arpa**

```
zone "2.2.10.in-addr.arpa" {  
    type master;  
    file "/home/core/primario/db.2-2-10.rev";  
    allow-transfer {10.2.2.3};  
};
```

Figure 29: *primario/named.conf.options* zona: **2.2.10.in-addr.arpa**


```

zone "3.3.10.in-addr.arpa" {
    type master;
    file "/home/core/primario/db.3-3-10.rev";
    allow-transfer {10.2.2.3;};
};

```

Figure 30: *primario/named.conf.options* zona: **3.3.10.in-addr.arpa**

```

zone "4.4.10.in-addr.arpa" {
    type master;
    file "/home/core/primario/db.4-4-10.rev";
    allow-transfer {10.2.2.3;};
};

```

Figure 31: *primario/named.conf.options* zona: **4.4.10.in-addr.arpa**

4)Baseando-se por exemplo no conteúdo do ficheiro *primario/db.local*, procure criar o ficheiro de dados do domínio de nomes: *primario/db.cc.pt* (incluir a informação de acordo com as regras definidas acima).

Para configurar este ficheiro tivemos de perceber o formato e sintaxe do *db.local* . Depois de uma breve e precisa pesquisa achamos que estavam preparados para configurar o mesmo. Assim o primeiro passo a tomar era configurar o nosso SOA (Start of Authority). Para isso tínhamos de escolher o servidor principal para esta zona, tendo sido nossa opção *dns.cc.pt.*. Depois para respeitar os requisitos previamente apresentados tivemos de usar o *grupo63.cc.pt* como administrador. Podemos ver essa configuração na seguinte imagem:

```

@      IN      SOA      dns.cc.pt. grupo63.cc.pt. (
                                1          ; Serial
                                604800     ; Refresh
                                86400      ; Retry
                                2419200    ; Expire
                                604800 )   ; Negative Cache TTL
.

```

Figure 32: SOA de *db.cc.pt*

Depois do SOA estar configurado era hora de passarmos aos *nameservers* e *mail*. Para definirmos os primeiros teríamos de usar a cláusula **NS** enquanto que para o mail a cláusula seria **MX**. Quanto ao mail é necessário ter em consideração que para além da cláusula e o endereço IP é necessário um valor inteiro que representa a prioridade. Esse valor quanto menor for, maior a sua prioridade. Além disto tendo o enunciado em consideração chegamos à seguinte configuração:

@	IN	NS	dns.cc.pt.
@	IN	NS	dns2.cc.pt.
@	IN	MX	10 mail.cc.pt.
@	IN	MX	20 Servidor2.cc.pt.

Figure 33: *NS e mail de db.cc.pt*

Passado isto era hora de inserir os registos e os respetivos endereços. Para isso usaríamos a cláusula **A** que nos permitia definir isso mesmo. Mais uma vez tendo em atenção os requisitos obtemos esta configuração:

mail	IN	A	10.1.1.3
www	IN	A	10.1.1.3
dns	IN	A	10.1.1.1
dns2	IN	A	10.2.2.3
Alfa	IN	A	10.3.3.1
Beta	IN	A	10.3.3.2
Gama	IN	A	10.3.3.3
Urano	IN	A	10.2.2.3
Neptuno	IN	A	10.2.2.2
Plutao	IN	A	10.2.2.1
Servidor1	IN	A	10.1.1.1
Servidor2	IN	A	10.1.1.2
Servidor3	IN	A	10.1.1.3
Cliente3	IN	A	10.4.4.3
Cliente2	IN	A	10.4.4.2
Cliente1	IN	A	10.4.4.1

Figure 34

Por fim foi hora de definir os *alias* desejados. Assim foi necessário usar a cláusula CNAME e obtivemos o seguinte resultado:

Grupo63	IN	CNAME	Cliente1
pop	IN	CNAME	Servidor2
imap	IN	CNAME	Servidor2

Figure 35

5) Baseando-se por exemplo no conteúdo do ficheiro `primario/db.127` procure criar o ficheiro de dados do(s) domínio(s) de reverse: `primario/db.1-1-10.rev` (ou outros, de acordo com as regras definidas acima); NOTA: o símbolo “@” é uma abreviatura do domínio que o ficheiro contém (ex: `cc.pt`); os nomes que não terminam com “.” são considerados relativos ao domínio do ficheiro; `www.cc.pt` sem o ponto é na verdade `www.cc.pt.cc.pt` .

Este ponto baseia-se em definir os domínios reverso. Este processo era muito semelhante eles, diferenciando apenas nos valores da interface de rede e o seu nome na topologia core. Para este tipo de ficheiros também teremos de inserir os *nameservers* . Depois todos os dados inseridos tem de seguir o mesmo formato: **Valores de interface de rede** IN PTR *nome na topologia.cc.pt.*. A principal diferença aqui é que iremos para o respetivo ficheiro apenas adicionar aqueles que pertencem a essa LAN. Assim as quatro imagens seguintes irão mostrar como ficaram definidos os ficheiros dos domínios reversos. De realçar que o SOA, administrador e *nameservers* permanecem iguais aos de *db.cc.pt* :

3	IN	PTR	mail.cc.pt.
1	IN	PTR	Servidor1.cc.pt.
2	IN	PTR	Servidor2.cc.pt.
3	IN	PTR	Servidor3.cc.pt.
3	IN	PTR	www.cc.pt.
1	IN	PTR	dns.cc.pt.
2	IN	PTR	pop.cc.pt.
2	IN	PTR	imap.cc.pt.

Figure 36: `db.1-1-10.rev`

1	IN	PTR	Plutao.cc.pt.
2	IN	PTR	Neptuno.cc.pt.
3	IN	PTR	dn2.cc.pt.
3	IN	PTR	Urano.cc.pt.

Figure 37: `db.2-2-10.rev`

1	IN	PTR	Alfa.cc.pt.
2	IN	PTR	Beta.cc.pt.
3	IN	PTR	Gama.cc.pt.

Figure 38: `db.3-3-10.rev`

1	IN	PTR	Cliente1.cc.pt.
2	IN	PTR	Cliente2.cc.pt.
3	IN	PTR	Cliente3.cc.pt.
1	IN	PTR	Grupo63.cc.pt.

Figure 39: db.4-4-10.rev

Configuração do servidor secundário

1) Editar o ficheiro secundario/named.conf.options por forma a incluir os servidores 193.136.9.240 e 193.136.19.1 (servidores do DI) como forwarders.

Para este passo fizemos o mesmo procedimento que fizemos para o primário.

```
forwarders {
    193.136.9.240;
    193.136.19.1;
};
```

Figure 40: secundario *named.conf.options*

2) Editar o ficheiro secundario/named.conf para incluir a indicação das novas zonas “cc.pt”, “1.1.10.in-addr.arpa” etc, mas desta vez apenas como zonas do tipo “slave” (ver manual ou exemplos). Não se esqueça de cláusula “masters” adequada. Assegure-se que os ficheiros de dados das zonas vão para /var/cache/bind/...db.cc.pt, db.1-1-10.rev, etc. por causa das permissões de escrita!

De forma a podermos garantir o total funcionamento dos servidores tivemos de alterar este ficheiro. Contrariamente ao servidor primário agora não estamos a falar dos servidores principais do DNS , mas sim os "escravos" dele. Assim as zonas vão ter agora type *slave* cujo seu masters será o IP do servidor principal 10.1.1.1 . De forma a respeitar o enunciado os ficheiros de dados das zonas irão para /var/cache/bind/ . Quanto ao número de zonas a abordagem do primário mantém-se. Assim nas seguintes imagens poderemos ver as diferentes zonas:

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

Figure 41: *primario/named.conf.options* zona: **cc.pt**

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

Figure 42: *primario/named.conf.options* zona: **1.1.10.in-addr.arpa**

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

Figure 43: *secundario/named.conf.options* zona: **2.2.10.in-addr.arpa**

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

Figure 44: *secundario/named.conf.options* zona: **3.3.10.in-addr.arpa**

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

Figure 45: *secundario/named.conf.options* zona: **4.4.10.in-addr.arpa**

Conclusão

Dado por terminado este terceiro trabalho prático o grupo acha que conseguiu corresponder às expectativas e tem consciência da importância deste trabalho para poder consolidar toda a matéria lecionada na Unidade Curricular de Comunicação por Computadores.

Assim o grupo está convencido que entregou uma implementação correta e funcional dos dois servidores e que apesar das dificuldades que lhes foram apresentadas, conseguimos supera-las e continuar todo o processo.