



Universidade do Minho

**Escola de Engenharia**

Mestrado Integrado em Engenharia Biomédica

**Serviços e Segurança em Redes Informáticas**

## **TP8 –Establish DNS service**



Trabalho elaborado pelo grupo:

Lara Alexandra Pereira Novo Martins Vaz (A88362)

Mariana Lindo Carvalho (A88360)

Tiago Miguel Parente Novais (A88397)

Braga, 18 de janeiro de 2022

---

## Índice

1.	Introdução.....	2
1.1	DNS.....	2
1.2	<i>Resource Records</i> .....	2
1.3	BIND.....	4
1.4	Objetivos .....	4
2.	Making use of the BIND package binaries, configure and setup a DNS service (with MASTER and SLAVE servers) for the domain:.....	5
2.1	Master files.....	5
2.1.1	named.conf.local.....	5
2.1.2	forward.MieBiom-g2.gcom.di.uminho.....	6
2.1.3	reverse.MieBiom-g2.gcom.di.uminho.....	8
2.2	Server file – named.conf.local.....	9
3.	Inicialização, reinicialização, terminação e status do servidor .....	10
4.	<i>Queries</i> formuladas ao DNS server desenvolvido e respetivos resultados.....	11
	Fontes.....	15

# 1. Introdução

## 1.1 DNS

O serviço de DNS (*Domain Name System*) é um dos serviços mais utilizados, fazendo parte do núcleo da Internet. Os servidores DNS são os responsáveis por traduzir nomes de *hosts* para endereços de IP e vice-versa <sup>[1]</sup>.

## 1.2 Resource Records

Um *Resource Record*, comumente designado de RR é a unidade de entrada de informações nos ficheiros de zona. Estes fornecem mapeamentos de nomes de hospedeiros para endereços IP, sendo utilizados para resolver todas as consultas ao DNS <sup>[2]</sup>.

Um RR é constituído por um tuplo de quatro elementos que contém os seguintes campos: (*Name, Value, Type, TTL*)

*TTL* é o tempo de vida útil do RR e determina quando um recurso deve ser removido de um cache. Os significados de *Name* e *Value* dependem do *Type*:

- Se *Type=A (Address)*, então *Name* é um nome de hospedeiro e *Value* é o endereço IPv4 para o nome de hospedeiro. Assim, um registo *Type A* fornece o mapeamento padrão entre nomes de hospedeiros e endereços IP <sup>[2]</sup>;
- Se *Type=NS (Name Server)*, então *Name* é um domínio e *Value* é o nome de um servidor DNS autoritário que sabe como obter os endereços IP para hospedeiros do domínio. Esse registo é usado para encaminhar consultas DNS ao longo da cadeia de consultas <sup>[2]</sup>;
- Se *Type=CNAME (Canonical Name)*, então *Value* é um nome canónico de hospedeiro para o apelido de hospedeiro contido em *Name*. Esse registo pode fornecer aos hospedeiros consultantes o nome canónico correspondente a um apelido de hospedeiro <sup>[2]</sup>;
- Se *Type=MX (Mail Exchange)*, então *Value* é o nome canónico de um servidor de mail cujo apelido de hospedeiro está contido em *Name*. Os RR MX permitem que os nomes de hospedeiros de servidores de mail tenham apelidos simples <sup>[2]</sup>;
- Se *Type=AAAA (IPv6 Address)*, então *Name* é um nome de hospedeiro e *Value* é o endereço IPv6 para o nome de hospedeiro. O RR AAAA é muito parecido ao Registro A, mas permite especificar o endereço IPv6 do servidor, em vez do IPv4 <sup>[3]</sup>.

Além destes RR existem ainda outros, como o PTR (*Pointer Record*). Enquanto que os registos DNS A são armazenados sob o nome de domínio fornecido, os registos DNS PTR são armazenados sob o endereço IP - invertido e com ".in-addr.arpa" no fim. "in-addr.arpa" deve ser adicionado porque os registos PTR são armazenados no domínio de nível superior .arpa no DNS, em que .arpa é um domínio usado principalmente para gerir a infraestrutura de rede. in-addr.arpa é o "namespace" dentro de .arpa para pesquisas de DNS reverso no IPv4. Os endereços IPv6 são construídos de forma diferente dos endereços IPv4 e os registos PTR IPv6 existem num "namespace" diferente dentro de .arpa. Os registos PTR IPv6 são armazenados sob o endereço IPv6, revertidos e convertidos em seções de quatro bits (em oposição a seções de 8 bits, como no IPv4), mais ". ip6.arpa" [4].

Por último, um outro RR muito importante e utilizado é o SOA (*Start Of Authority*), que armazena informações importantes sobre um domínio ou zona, como o endereço de e-mail do administrador, quando o domínio foi atualizado pela última vez e quanto tempo o servidor deve esperar entre as atualizações. Para definir um RR do tipo SOA, é necessário ter em conta o seu formato, que inclui os seguintes parâmetros: *Serial number*, *Refresh interval*, *Retry interval*, *Expire time*, *Negative Cache TTL* [5].

Relativamente ao *Serial Number*, este é utilizado como um número de versão para a zona DNS. Assim sendo, sempre que a configuração do domínio sofrer qualquer alteração, o *Serial Number* do Master é incrementado e o *Slave* é informado de tal alteração. É de notar que este número deve ser alterado para que seja possível ao *Slave* atualizar os seus dados, havendo assim entendimento entre o *Slave* e o Master. Posto isto, o *Serial Number* tem o seguinte formato: *yyyymmddnn* [5].

- *yyyy*: ano;
- *mm*: mês;
- *dd*: dia ;
- *nn*: número de alterações feitas nesse dia.

Quanto ao *Refresh interval*, este corresponde à frequência com que o *Slave* pesquisa o *Serial Number* do Master, para verificar os novos dados. O *Retry interval* diz respeito à frequência com que o *Slave* verifica o *Serial Number*, caso o Master não tenha respondido. Relativamente ao *Expire time*, este identifica o intervalo de tempo em que se o *Slave* for incapaz de contactar o Master, então o *Slave* vai apagar a sua cópia do ficheiro da zona. O *Negative Cache TTL* é usado para armazenamento negativo em cache e indica o tempo que a cache pode armazenar a inexistência de um RR [5].

### 1.3 BIND

O BIND (*Berkeley Internet Name Domain*) é conjunto de ferramentas de software que inclui o software do servidor de nomes mais usado no mundo, o DNS. O BIND pode ser usado para executar um servidor DNS de cache ou um servidor de nomes autoritário e fornece recursos como balanceamento de carga, notificação, atualização dinâmica, DNS dividido, DNSSEC, IPv6 e muitos outros <sup>[1]</sup>.

Se por um lado o DNS especifica o processo pelo qual um computador consegue encontrar outro, o BIND contém todos os softwares necessários para fazer questões e responder aquelas colocadas ao serviço. O software BIND é constituído pelo *Domain Name Resolver*, que consiste na resolução de questões relacionadas com nomes, redirecionando-as para os servidores corretos de forma a lhes dar resposta. Os vários *resolvers* estabelecem ligações entre si até saberem o endereço IP a que aquele DNS pertence. Estas ligações podem ser compostas por perguntas enviadas a um ou vários servidores autoritários. O BIND é também constituído por um *Domain Name Authority Server* que responde aos pedidos vindos dos *resolvers* anteriores, baseando-se nos nomes dos domínios pelos quais são autoritários. Um outro componente muito importante do BIND são as ferramentas usualmente usadas num terminal, como o *nslookup*, que poderão ser utilizadas no BIND com um qualquer servidor de DNS <sup>[1]</sup>.

### 1.4 Objetivos

Para a realização deste trabalho, foi necessário instalar o BIND nos computadores dos elementos do grupo. O objetivo deste trabalho é configurar um servidor DNS com servidores do tipo Master e *Slave*, para o domínio MieBiom-g2.gcom.di.uminho.pt. no qual, 2 corresponde ao número do grupo.

## 2. Making use of the BIND package binaries, configure and setup a DNS service (with MASTER and SLAVE servers) for the domain:

MieBiom-gX.gcom.di.uminho.pt.

where X is your group number, using a minimal configuration setup. Minimal configuration must include: SOA records, NS records, A records, PTR records, AAAA records and MX records.

### 2.1 Master files

#### 2.1.1 named.conf.local

Inicialmente, começou-se por definir o ficheiro `named.conf.local`, em que `/usr/local/etc/named.conf.local` aponta para o ficheiro de zona, criado manualmente, que contém os RR especificados no enunciado.

Para a configuração do ficheiro `named.conf.local`, foi necessário identificar de dois modos. O primeiro modo corresponde ao *Forward Mode* (Figura 1) que permite resolver nomes de *hosts* nos endereços IP correspondentes. Neste modo, foi primeiramente definida uma zona, zone “MieBiom-g2.com.di.uminho.pt.”, cujo tipo de servidor é Master e o endereço de IP do *Slave* para o qual é permitida a transferência do ficheiro de zona é 192.168.1.2. (*allow-transfer {192.168.1.2;}*). Foi também necessária a indicação da localização do respetivo ficheiro, que neste caso é `/etc/bind/forward.MieBiom-g2.gcom.di.uminho.pt`. Verifica-se também que não é permitido a nenhum *Slave* alterar o ficheiro de zona (*allow-update {none;}*) e que cada vez que esse ficheiro é alterado, o *Slave*, cujo endereço IP é 192.168.1.2, é notificado da alteração (*also-notify {192.168.1.2;}*). Essa notificação permite que o *Slave* possa atualizar a sua cópia mais rapidamente, em vez de ter de esperar pelo *Refresh Interval* para verificar se a sua cópia está atualizada.

O segundo modo definido no ficheiro `named.conf.local` é o *Reverse Mode* (Figura 2), que permite resolver endereços de IP em nomes de *hosts*. Do mesmo modo que no *Forward Mode*, no *Reverse Mode*, também é definida uma zona, que neste caso é “1.168.192.in-addr.arpa”, em que o domínio “in-addr.arpa” é um domínio especial reservado para a tradução de endereços IP em nomes de *hosts*. O tipo de servidor é também Master e o endereço de IP do *Slave* para o qual é permitida a transferência do ficheiro de zona é 192.168.1.2. (*allow-transfer {192.168.1.2;}*). Foi

também necessária a indicação da localização do respetivo ficheiro, que neste caso é `"/etc/bind/reverse.MieBiom-g2.gcom.di.uminho.pt"`. Verifica-se também que não é permitido a nenhum *Slave* alterar o ficheiro de zona (`allow-update {none}`) e que cada vez que esse ficheiro é alterado, o *Slave*, cujo endereço IP é 192.168.1.2, é notificado da alteração (`also-notify {192.168.1.2}`).

```
1 // Forward Mode
2 zone "MieBiom-g2.gcom.di.uminho.pt" IN {
3     type master;
4     file "/etc/bind/forward.MieBiom-g2.gcom.di.uminho.pt";
5     allow-update { none; };
6     allow-transfer { 192.168.1.2; };
7     also-notify { 192.168.1.2; };
8 };
```

**Figura 1.** Forward Mode presente no ficheiro `named.conf.local` do Master.

```
// Reverse mode
zone "1.168.192.in-addr.arpa" IN {
    type master;
    file "/etc/bind/reverse.MieBiom-g2.gcom.di.uminho.pt";
    allow-update { none; };
    allow-transfer { 192.168.1.2; };
    also-notify { 192.168.1.2; };
};
```

**Figura 2.** Reverse Mode presente no ficheiro `named.conf.local` do Master.

### 2.1.2 forward.MieBiom-g2.gcom.di.uminho

Após a criação de uma zona, é necessário adicionar RR. No caso do ficheiro `forward.MieBiomg2.gcom.di.uminho`, representado na Figura 3, estão definidos os RR que são responsáveis por responder às *queries* no *Forward Mode*. Inicialmente, começou-se por definir um TTL de 604800s, utilizado por defeito caso não seja especificado um outro valor. De seguida, definiu-se um RR do tipo SOA, cujo nome do *hostname* do servidor de nomes Master é `ns1.MieBiom-g2.gcom.di.uminho.pt`. e o endereço de email da pessoa responsável, sendo o `@` substituído pelo ponto e ponto final é `root.MieBiom-g2.gcom.di.uminho.pt`. O *Serial Number* é 3, não tendo o formato habitual de um *Serial Number*, correspondendo apenas ao número de vezes que o ficheiro foi alterado. O *Refresh Interval* é 604800s, o *Retry Interval* é 86400, o *Expire Time* é 2419200 e o *Negative Cache TTL* é 604800, que por sua vez é igual ao valor de TTL usado por defeito.

De seguida, foram definidos outros RR, como:

- Dois registos do tipo NS, o ns1: ns1.MieBiom-g2.gcom.di.uminho.pt. e o ns2.MieBiom-g2.gcom.di.uminho.pt;
- Dois registos do tipo A, contém os endereços IPv4 do ns1 (servidor Master) e ns2 (servidor *Slave*) que são 192.168.1.80 e 192.168.18.2, respetivamente;
- Dois registos do tipo AAAA que contém os endereços IPv6 do ns1 (servidor Master) e ns2 (servidor *Slave*) que são fe80::f48f:55c6:cba5:4b44 e fe80::bc05:bbd7:55ae:7382, respetivamente;
- Um registo do tipo MX, cujo *host* do servidor de mail do domínio é mx1, ou seja, mx1.MieBiom-g2.gcom.di.uminho.pt. e a prioridade é 10;
- Dois registos do tipo A que contém os endereços de IPv4 dos servidores apps1.MieBiom-g2.gcom.di.uminho.pt. e mx1. MieBiom-g2.gcom.di.uminho.pt. que são 192.168.1.14 e 192.168.1.15, respetivamente;
- Dois registos do tipo CNAME, cujos nomes alternativos ao nome canónico apps1.MieBiom-g2.gcom.di.uminho.pt. são www.MieBiom-g2.gcom.di.uminho.pt. e ftp.MieBiom-g2.gcom.di.uminho.pt.

```
1 $TTL 604800
2 @ IN SOA ns1.MieBiom-g2.gcom.di.uminho.pt. root.MieBiom-g2.gcom.di.uminho.pt. (
3     3 ; Serial
4     604800 ; Refresh
5     86400 ; Retry
6     2419200 ; Expire
7     604800 ) ; Negative Cache TTL
8
9 ;--- Name Server Information
10 @ IN NS ns1.MieBiom-g2.gcom.di.uminho.pt.
11 @ IN NS ns2.MieBiom-g2.gcom.di.uminho.pt.
12
13 ;--- IP address of Name Server
14 ns1 IN A 192.168.1.80
15 ns2 IN A 192.168.18.2
16
17 ; IPv6 addresses of named servers
18 ns1 IN AAAA fe80::f48f:55c6:cba5:4b44
19 ns2 IN AAAA fe80::bc05:bbd7:55ae:7382
20
21 ;--- Mail Exchanger ( if exists )
22 MieBiom-g2.gcom.di.uminho.pt. IN MX 10 mx1.
23
24 ;--- A - Record HostName To Ip Address
25 apps1 IN A 192.168.1.14
26 mx1 IN A 192.168.1.15
27
28 ;--- CNAME record
29 www IN CNAME apps1.MieBiom-g2.gcom.di.uminho.pt.
30 ftp IN CNAME apps1.MieBiom-g2.gcom.di.uminho.pt.
```

**Figura 3.** Conteúdo do ficheiro forward.MieBiom-g2.gcom.di.uminho.



### 2.1.3 reverse.MieBiom-g2.gcom.di.uminho

Neste ficheiro, representado na Figura 4, estão definidos os RR que são responsáveis por responder às queries no *Reverse Mode*. Inicialmente, começou-se por definir um TTL de 604800s, utilizado por defeito caso não seja especificado um outro valor. De seguida, definiu-se um RR do tipo SOA, cujo nome do *hostname* do servidor de nomes Master é MieBiom-g2.gcom.di.uminho.pt. e o endereço de email da pessoa responsável, sendo o @ substituído pelo ponto e ponto final é root.MieBiom-g2.gcom.di.uminho.pt. O *Serial Number* é 3, não tendo o formato habitual de um *Serial Number*, correspondendo apenas ao número de vezes que o ficheiro foi alterado. O *Refresh Interval* é 604800s, o *Retry Interval* é 86400, o *Expire Time* é 2419200 e o *Negative Cache TTL* é 604800, que por sua vez é igual ao valor de TTL usado por defeito.

De seguida, foram definidos outros RR, como:

- Dois registos do tipo NS, o ns1 (servidor Master): ns1.MieBiom-g2.gcom.di.uminho.pt. e o ns2 (servidor *Slave*): ns2.MieBiom-g2.gcom.di.uminho.pt;
- Dois registos do tipo PTR que permitem que quando se façam pesquisas inversas (pesquisar pelo IPv4 em vez de pesquisar pelo nome) seja possível associar aos endereços IPv4 192.168.1.80 e 192.168.1.2 os servidores ns1.MieBiom-g2.gcom.di.uminho.pt. e ns2.MieBiom-g2.gcom.di.uminho.pt., respetivamente;
- Dois registos do tipo PTR que permitem que quando se façam pesquisas inversas seja possível associar aos endereços IPv4 192.168.1.14 e 192.168.1.15 os servidores apps1.MieBiom-g2.gcom.di.uminho.pt. e mail1.MieBiom-g2.gcom.di.uminho.pt., respetivamente.

```
$TTL 604800
@ IN SOA MieBiom-g2.gcom.di.uminho.pt. root.MieBiom-g2.gcom.di.uminho.pt. (
    3 ; Serial
    604800 ; Refresh
    86400 ; Retry
    2419200 ; Expire
    604800 ); Negative Cache TTL

;--Name Server Information
@ IN NS ns1.MieBiom-g2.gcom.di.uminho.pt.
@ IN NS ns2.MieBiom-g2.gcom.di.uminho.pt.

;--Reverse lookup for Name Server
80 IN PTR ns1.MieBiom-g2.gcom.di.uminho.pt.
2 IN PTR ns2.MieBiom-g2.gcom.di.uminho.pt.

;--PTR Record IP address to HostName
14 IN PTR apps1.MieBiom-g2.gcom.di.uminho.pt.
15 IN PTR mail1.MieBiom-g2.gcom.di.uminho.pt.
```

**Figura 4.** Conteúdo do ficheiro reverse.MieBiom-g2.gcom.di.uminho.

## 2.2 Server file – named.conf.local

Para a configuração do ficheiro named.conf.local do *Slave*, foi necessário identificar de dois modos. O primeiro modo corresponde ao *Forward Mode* (Figura 5) que permite resolver nomes de *hosts* nos endereços IP correspondentes. Neste modo, foi primeiramente definida uma zona, zone “MieBiom-g2.com.di.uminho.pt.”, cujo tipo de servidor é *Slave* e o endereço de IP do Master para o qual o *Slave* pode transferir cópias do ficheiro de zona é 192.168.1.80. (*masters {192.168.1.80;}*). Foi também necessária a indicação da localização do respetivo ficheiro de zona, que neste caso é “forward.MieBiom-g2.gcom.di.uminho.pt”.

O segundo modo definido no ficheiro named.conf.local do *Slave* é o *Reverse Mode* (Figura 6), que permite resolver endereços de IP em nomes de *hosts*. Do mesmo modo que no *Forward Mode*, no *Reverse Mode*, também é definida uma zona, que neste caso é “1.168.192.in-addr.arpa”, em que o domínio “in-addr.arpa” é um domínio especial reservado para a tradução de endereços IP em nomes de *hosts*. O tipo de servidor é também *Slave* e o endereço de IP do Master para o qual o *Slave* pode transferir o ficheiro de zona é 192.168.1.80. (*masters {192.168.1.80;}*). Foi também necessária a indicação da localização do respetivo ficheiro, que neste caso é “reverse.MieBiom-g2.gcom.di.uminho.pt”.

```
//Forward Zone
zone "MieBiom-g2.gcom.di.uminho.pt" IN { //Domain name
    type slave; //Secondary Slave dns
    file "forward.MieBiom-g2.gcom.di.uminho.pt"; //Forward Zone Cache file
    masters { 192.168.1.80; }; //Master Server IP
};
```

**Figura 5.** *Forward Mode* presente no ficheiro named.conf.local do *Slave*.

```
//Reverse Zone
zone "1.168.192.in-addr.arpa" IN {
    type slave; // Secondary/Slave dns
    file "reverse.MieBiom-g2.gcom.di.uminho.pt"; //Reverse Zone Cache file
    masters { 192.168.1.80; }; //Master Server IP
};
```

**Figura 6.** *Reverse Mode* presente no ficheiro named.conf.local do *Slave*.

### 3. Inicialização, reinicialização, terminação e status do servidor

De forma a inicializar o servidor desenvolvido para ser possível fazer consultas, foi utilizado o comando `service bind9 start`. Para reinicializar o servidor, foi utilizado o comando `service bind9 restart` e para terminar o servidor utilizou-se o comando `service bind9 stop`. Foi ainda utilizado o comando `service bind9 status` para verificar se o servidor estava a correr corretamente. Nas Figuras 7 a 10 estão apresentados os comandos referidos e na Figura 11 está representado o resultado obtido após utilizar o comando `service bind9 status`, em que é visível que o servidor está a correr normalmente.

```
mariana@mariana-VirtualBox:/etc/bind$ service bind9 start
```

Figura 7. Comando utilizado para inicializar o servidor.

```
mariana@mariana-VirtualBox:/etc/bind$ service bind9 restart
```

Figura 8. Comando utilizado para reinicializar o servidor.

```
mariana@mariana-VirtualBox:/etc/bind$ service bind9 stop
```

Figura 9. Comando utilizado para terminar o servidor.

```
mariana@mariana-VirtualBox:/etc/bind$ service bind9 status
```

Figura 10. Comando utilizado para averiguar o status do servidor.

```
● named.service - BIND Domain Name Server
   Loaded: loaded (/lib/systemd/system/named.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2022-01-17 16:35:48 WET; 3s ago
     Docs: man:named(8)
    Main PID: 7550 (named)
      Tasks: 14 (limit: 9469)
     Memory: 28.3M
    CGroup: /system.slice/named.service
            └─7550 /usr/sbin/named -f -u bind

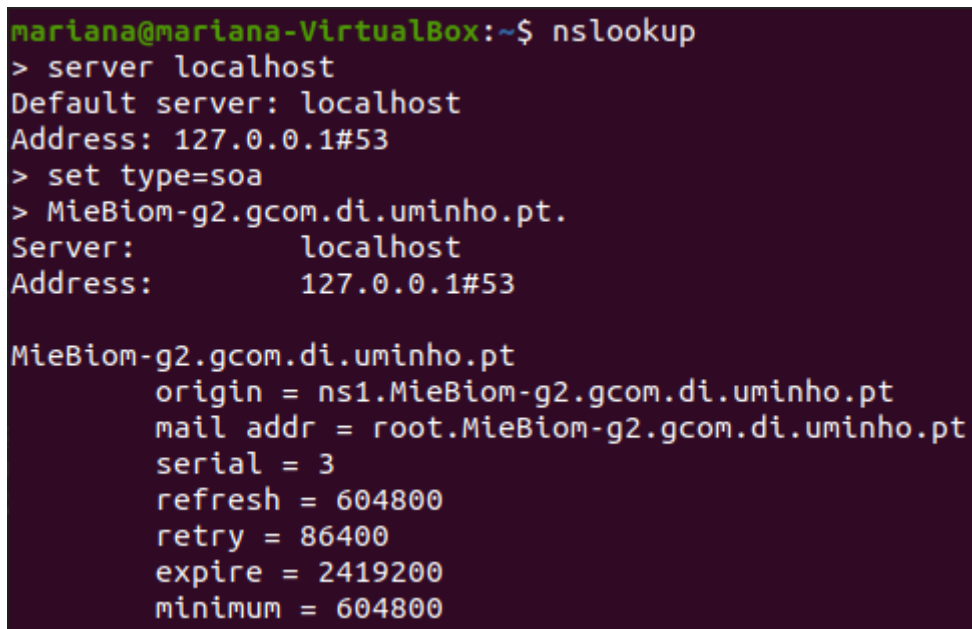
jan 17 16:35:48 mariana-VirtualBox named[7550]: network unreachable resolving './NS/IN': 2001:500:a8::e#53
jan 17 16:35:48 mariana-VirtualBox named[7550]: network unreachable resolving './DNSKEY/IN': 2001:500:12::d0d#53
jan 17 16:35:48 mariana-VirtualBox named[7550]: network unreachable resolving './NS/IN': 2001:500:12::d0d#53
jan 17 16:35:48 mariana-VirtualBox named[7550]: network unreachable resolving './DNSKEY/IN': 2001:503:c27::2:30#53
jan 17 16:35:48 mariana-VirtualBox named[7550]: network unreachable resolving './NS/IN': 2001:503:c27::2:30#53
jan 17 16:35:49 mariana-VirtualBox named[7550]: network unreachable resolving './DNSKEY/IN': 2001:500:12::d0d#53
jan 17 16:35:49 mariana-VirtualBox named[7550]: network unreachable resolving './DNSKEY/IN': 2001:500:200::b#53
jan 17 16:35:49 mariana-VirtualBox named[7550]: network unreachable resolving './DNSKEY/IN': 2001:7fe::53#53
jan 17 16:35:49 mariana-VirtualBox named[7550]: managed-keys-zone: Key 20326 for zone . is now trusted (acceptance timer complete)
jan 17 16:35:49 mariana-VirtualBox named[7550]: resolver priming query complete
```

Figura 11. Resultado do comando `service bind9 status`.

#### 4. *Queries* formuladas ao DNS server desenvolvido e respetivos resultados

- **SOA record**

Estando o servidor ligado (com o comando presente nas Figuras 7 ou 8), é possível formular algumas *queries* ao Master. Para isso utilizou-se o comando *nslookup* e especificou-se que o server era o *localhost*. De seguida, utilizou-se o comando *set type=soa* para obter respostas autoritárias do domínio MieBiom-g2.gcom.di.uminho.pt., como representado na Figura 12. Analisando essa figura, verifica-se que estão identificados o *origin* que é o Master, o *mail addr* que é o especificado nos ficheiros forward.MieBiom-g2.gcom.di.uminho e forward.MieBiom-g2.gcom.di.uminho e todos os restantes parâmetros pertencentes ao RR do tipo SOA, igualmente identificados no ficheiro reverse.MieBiom-g2.gcom.di.uminho.



```
mariana@mariana-VirtualBox:~$ nslookup
> server localhost
Default server: localhost
Address: 127.0.0.1#53
> set type=soa
> MieBiom-g2.gcom.di.uminho.pt.
Server:          localhost
Address:         127.0.0.1#53

MieBiom-g2.gcom.di.uminho.pt
    origin = ns1.MieBiom-g2.gcom.di.uminho.pt
    mail addr = root.MieBiom-g2.gcom.di.uminho.pt
    serial = 3
    refresh = 604800
    retry = 86400
    expire = 2419200
    minimum = 604800
```

**Figura 12.** Resposta obtida ao efetuar a *querie nslookup set type=soa MieBiom-g2.gcom.di.uminho.pt.*

- **NS record**

Para obter RR do tipo NS, recorreu-se ao comando *nslookup set type=ns*, obtendo-se assim os servidores de nomes do domínio em estudo, conforme representado na Figura 13. Analisando a figura, verifica-se que estão identificados os servidores de nomes para o domínio em causa e são exatamente os que estão definidos no ficheiros forward.MieBiom-g2.gcom.di.uminho e reverse.MieBiom-g2.gcom.di.uminho.

```

> set type=ns
> MieBiom-g2.gcom.di.uminho.pt.
Server:      localhost
Address:     127.0.0.1#53

MieBiom-g2.gcom.di.uminho.pt      nameserver = ns2.MieBiom-g2.gcom.di.uminho.pt.
MieBiom-g2.gcom.di.uminho.pt      nameserver = ns1.MieBiom-g2.gcom.di.uminho.pt.

```

Figura 13. Resposta obtida ao efetuar a *querie nslookup set type=ns MieBiom-g2.gcom.di.uminho.pt.*

- **A record**

Para obter RR do tipo A, recorreu-se ao comando *nslookup set type=a*, obtendo-se assim os endereços IPv4 dos servidores de nomes do domínio em estudo, conforme representado na Figura 14. Analisando a figura, verifica-se que estão identificados os endereços IPv4 dos servidores de nomes para o domínio em causa e são exatamente os que estão definidos no ficheiro forward.MieBiom-g2.gcom.di.uminho. É de realçar que para os servidores de nomes alternativos *www.MieBiom-g2.gcom.di.uminho.* e *ftp.MieBiom-g2.gcom.di.uminho.* é ainda referido o nome canónico que é *apps1.MieBiom-g2.gcom.di.uminho.*, pois os 2 primeiros são nomes alternativos deste último, mas não deixam de representar o mesmo servidor, e por isso, apresentam o mesmo endereço IP.

```

> set type=a
> ns1.MieBiom-g2.gcom.di.uminho.pt.
Server:      localhost
Address:     127.0.0.1#53

Name:  ns1.MieBiom-g2.gcom.di.uminho.pt
Address: 192.168.1.80
> set type=a
> ns2.MieBiom-g2.gcom.di.uminho.pt.
Server:      localhost
Address:     127.0.0.1#53

Name:  ns2.MieBiom-g2.gcom.di.uminho.pt
Address: 192.168.18.2
> set type=a
> mx1.MieBiom-g2.gcom.di.uminho.pt.
Server:      localhost
Address:     127.0.0.1#53

Name:  mx1.MieBiom-g2.gcom.di.uminho.pt
Address: 192.168.1.15
> set type=a
> apps1.MieBiom-g2.gcom.di.uminho.pt.
Server:      localhost
Address:     127.0.0.1#53

Name:  apps1.MieBiom-g2.gcom.di.uminho.pt
Address: 192.168.1.14
> set type=a
> www.MieBiom-g2.gcom.di.uminho.pt.
Server:      localhost
Address:     127.0.0.1#53

www.MieBiom-g2.gcom.di.uminho.pt      canonical name = apps1.MieBiom-g2.gcom.di.uminho.pt.
Name:  apps1.MieBiom-g2.gcom.di.uminho.pt
Address: 192.168.1.14
> set type=a
> ftp.MieBiom-g2.gcom.di.uminho.pt.
Server:      localhost
Address:     127.0.0.1#53

ftp.MieBiom-g2.gcom.di.uminho.pt      canonical name = apps1.MieBiom-g2.gcom.di.uminho.pt.
Name:  apps1.MieBiom-g2.gcom.di.uminho.pt
Address: 192.168.1.14

```

Figura 14. Resposta obtida ao efetuar a *querie nslookup set type=a.*

- **AAAA record**

Para obter RR do tipo AAAA, recorreu-se ao comando *nslookup set type=aaaa*, obtendo-se assim os endereços IPv6 dos servidores de nomes do domínio em estudo, conforme representado na Figura 15. Analisando a figura, verifica-se que estão identificados os endereços IPv6 dos servidores de nomes para o domínio em causa e são exatamente os que estão definidos no ficheiro *forward.MieBiom-g2.gcom.di.uminho*.



```
> set type=aaaa
> ns1.MieBiom-g2.gcom.di.uminho.pt.
Server:      localhost
Address:     127.0.0.1#53

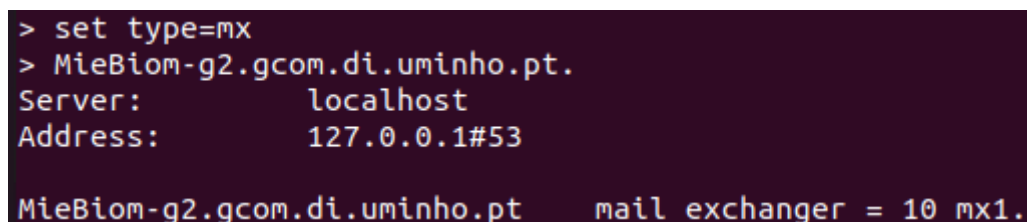
Name:   ns1.MieBiom-g2.gcom.di.uminho.pt
Address: fe80::f48f:55c6:cba5:4b44
> set type=aaaa
> ns2.MieBiom-g2.gcom.di.uminho.pt.
Server:      localhost
Address:     127.0.0.1#53

Name:   ns2.MieBiom-g2.gcom.di.uminho.pt
Address: fe80::bc05:bbd7:55ae:7382
```

Figura 15. Resposta obtida ao efetuar a *querie nslookup set type=aaaa*.

- **MX record**

Para obter RR do tipo MX, recorreu-se ao comando *nslookup set type=mx*, obtendo-se assim o destinatário de e-mail do domínio em estudo, conforme representado na Figura 16. Analisando a figura, verifica-se que está identificado o servidor de mail para o domínio em causa e é exatamente os que está definido no ficheiro *forward.MieBiom-g2.gcom.di.uminho*.



```
> set type=mx
> MieBiom-g2.gcom.di.uminho.pt.
Server:      localhost
Address:     127.0.0.1#53

MieBiom-g2.gcom.di.uminho.pt    mail exchanger = 10 mx1.
```

Figura 16. Resposta obtida ao efetuar a *querie nslookup set type=mx*.

- **PTR record**

De forma a exemplificar o funcionamento de um PTR record, em que se utiliza pesquisa inversa (*reverse*), foram formuladas *queries* do tipo *set type=ns* em que em vez de se pesquisar por um nome de servidor para obter o endereço IP, pesquisou-se pelo IPv4 e obteve-se os servidores de

nomes que correspondiam aos respectivos endereços IP colocados na questão. Isto está representado na Figura 17.

```
> set type=ns
> 192.168.1.80
Server:      localhost
Address:     127.0.0.1#53

80.1.168.192.in-addr.arpa      name = ns1.MieBiom-g2.gcom.di.uminho.pt.
> set type=ns
> 192.168.1.2
Server:      localhost
Address:     127.0.0.1#53

2.1.168.192.in-addr.arpa      name = ns2.MieBiom-g2.gcom.di.uminho.pt.
> set type=ns
> 192.168.1.14
Server:      localhost
Address:     127.0.0.1#53

14.1.168.192.in-addr.arpa     name = apps1.MieBiom-g2.gcom.di.uminho.pt.
> set type=ns
> 192.168.1.15
Server:      localhost
Address:     127.0.0.1#53

15.1.168.192.in-addr.arpa     name = mail1.MieBiom-g2.gcom.di.uminho.pt.
```

Figura 17. Resposta obtida ao efetuar a *querie nslookup up set type=ns*, mas com pesquisa inversa.

- **Slave**

Normalmente, o DNS *Slave* server serve sobretudo para melhorar questões de performance, "proteger" o servidor principal e para backup/redundância. Por esse motivo, o DNS *Slave* é colocado noutra máquina diferente do master. Deste modo, para verificar a coerência de dados entre o Master e o *Slave*, deveriam ser executados todos os comandos referidos anteriormente no Master e obter as mesmas respostas. Isso comprovaria a existência de uma boa ligação entre o Master e o *Slave*.

Contudo, devido a dificuldades técnicas, em que os computadores dos elementos do grupo não estavam na mesma rede, não foi possível testar o funcionamento do *Slave*.

## Fontes

- [1] A CLOUD GURU. <https://acloudguru.com/blog/engineering/all-about-bind-dns-who-how-why>. Acedido a 16 dez 2021.
- [2] F. Kurose and K. W. Ross, Redes de Computadores e a Internet. 2013.
- [3] Crazy domains. <https://www.crazydomains.com.au/help/what-is-an-aaaa-record/>. Acedido a 15 jan 2022.
- [4] Claude Flare. <https://www.cloudflare.com/learning/dns/dns-records/dns-ptr-record/>. Acedido a 15 jan 2022.
- [5] Claude Flare. <https://www.cloudflare.com/learning/dns/dns-records/dns-soa-record/>. Acedido a 15 jan 2022.