Universidade Do Minho

Mestrado Integrado Em Engenharia Biomédica

Comunicações e Redes

SERVIÇO DE DNS

Group Work

Braga, Dezembro 2012

Universidade Do Minho

Mestrado Integrado Em Engenharia Biomédica

Comunicações e Redes

SERVIÇO DE DNS

Group Work

Docente: Alexandre Santos

Grupo 4: Hugo Gomes (58536)
Marta Moreira (54459)
Telma Veloso (58521)

Resumo

Pretendia-se estabelecer um serviço de DNS envolvendo três máquinas distintas, respectivamente configuradas como:

1. *Master* do domínio directo `miebiom-g4.gcom.di.uminho.pt.` e *slave* do domínio reverso `2.168.192.in-addr.arpa.`;
2. *Master* do domínio reverso e *slave* do domínio directo;
3. *Slave* do domínio reverso.

Neste sentido, construíram-se ficheiros adaptados a cada um destes casos específicos, nomeadamente ficheiros de raiz, configuração e zona, com o intuito de definir as zonas suportadas por cada um dos servidores de nomes e caracterizá-las autoritativamente, entre outros aspectos. O referido sistema foi implementado recorrendo à versão `9.8.4-P1` do `BIND` e apresenta, naturalmente, algumas lacunas a nível de segurança, que poderiam ser melhoradas ou mesmo extintas através de medidas como a definição dos servidores em sub-redes distintas.

Conteúdo

1 Introdução

O desenvolvimento do sistema de DNS (Domain Name System) surgiu da necessidade premente de facultar aos utilizadores da *Internet* um serviço que contornasse uma realidade em que estes fossem forçados a decorar um número infindável de endereços de IP, de forma a poderem aceder às aplicações pretendidas — o que tornaria, obviamente, o funcionamento da rede muito pouco prático ou mesmo impossível. Assim, a definição de um serviço que, na sua base, faculta a tradução de *host names* em endereços de IP revela-se de importância e utilidade fulcrais, no que diz respeito à forma como a sociedade hodierna comunica.

De forma a simplificar a vertente administrativa, as especificações do DNS definem dois tipos de servidores de nomes: *primary masters* e *secondary masters*, estando os primeiros associados a servidores de nomes que lêem a informação de zona a partir de um ficheiro localizado no próprio *host* e os segundos a servidores que obtêm a informação de zona através de outro servidor, autoritativo para essa mesma zona — denominado de *master*. Normalmente, o *master* coincide com o *primary master* da zona, embora um *secondary master* — também denominado de *slave* — possa obter a informação de zona através de uma transferência de zona derivada de um contacto com o servidor que, perante este, desempenha funções de *master*, pelo que tanto o *primary master* como o *slave* são considerados autoritativos para essa zona [1].

No que concerne a segurança, os servidores de nomes expostos à Internet encontram-se sujeitos a uma grande variedade de ataques, entre os quais os ataques ao protocolo, que exploram as falhas na implementação do protocolo de DNS, isto é, a forma como o DNS funciona no seio da *Internet*. Consideram-se, geralmente, três tipos de ataque:

1. `DNS Spoofing`;
2. `DNS ID Hacking`;
3. `DNS Cache Poisoning`.

O `DNS Spoofing` diz respeito à acção de responder a um pedido de DNS originalmente direccionado a um outro servidor, não se estabelecendo diferença funcional entre situações de pedido de mapeamento servidor-servidor e cliente-servidor. O *hacker* faz o *spoof* da resposta respondendo com o endereço de IP do servidor real no campo *source-address* dos pacotes. Contudo, para que o *spoof* seja levado a cabo com sucesso, é necessário que o *hacker* tenha acesso ao número de identificação das *queries*, acção levada a cabo através do `DNS ID Hacking`. Torna-se, desta forma, possível personificar uma resposta DNS de forma a direccionar o cliente por um caminho diferente do pretendido, sem nunca, porém, interferir com a *cache* do servidor que está a ser personificado. No entanto, caso o cliente seja um outro servidor de DNS, a *cache* será inevitavelmente infectada, num processo denominado de `DNS Cache Poisoning`, que consiste num ataque que leva o servidor a armazenar informação falsa, forçando-o a responder a um pedido com a informação que o *hacker* pretende que seja dada como resposta. Os ataques que utilizam `DNS Cache Poisoning` são considerados, actualmente, ultrapassados, visto que as versões do `BIND` posteriores à sua descoberta foram sendo progressivamente optimizadas no sentido de preveni-los [2].

Um outro tipo de ataques, bastante mais actuais e com tendência de evolução exponencial, corresponde aos ataques ao servidor de DNS, que exploram, não a implementação defeituosa do protocolo em si, mas o próprio servidor. Distinguem-se, por norma, dois tipos de ataque, neste campo [2]:

1. Ataques que tiram partido de eventuais *bugs* que existam na implementação do *software* de DNS — como sejam as situações de *buffer overflow*, no `BIND` — ou mesmo noutro serviço em execução na

máquina do servidor de DNS;

2. Ataques de negação de serviço, por intermédio das técnicas de `Flooding` ou `ICMP Smurfing`, por exemplo.

De entre as medidas que podem ser tomadas para evitar (ou minorar) os efeitos dos ataques anteriormente referidos, contam-se:

1. Proibir *queries* recursivas, de forma a prevenir o *spoofing*;
2. Actualizar frequentemente a versão do `BIND` utilizada, para limitar os problemas de segurança relacionados com *bugs*;
3. Evitar manter a totalidade dos servidores de DNS na mesma sub-rede, no sentido de contrariar a existência de um ponto de falha único;
4. Restringir ao mínimo as *queries* possíveis e as permissões relacionadas com os *hosts* que podem efectuá-las.

Num perímetro exterior ao dos ataques encontra-se a questão das transferências de zona, que podem potencialmente fornecer a um *hacker* informação acerca da rede da vítima — como sejam o número de *hosts*, nomes e endereços — e identificar potenciais alvos futuros. Neste sentido, esta funcionalidade deve ser circunscrita exclusivamente aos *hosts* específicos que de facto a requeiram [2, 3].

$\nabla \nabla \nabla$

O presente relatório, elaborado no âmbito da Unidade Curricular de Comunicações e Redes, visa descrever o processo de desenvolvimento e implementação de um serviço de DNS para o domínio `miebiom-g4.gcom.di.uminho.pt.`, explicitando os ficheiros utilizados e apresentando uma breve discussão acerca das funcionalidades do serviço, bem como das características de segurança que lhe são inerentes.

2 Ficheiros

2.1 Máquina 1

A Máquina 1 foi configurada como *master* do domínio directo `miebiom-g4.gcom.di.uminho.pt` e *slave* do domínio `2.168.192.in-addr.arpa.`, gerido pela Máquina 2.

2.1.1 Ficheiro de Raiz

O Ficheiro de Raiz (*Root Zone File*) compreende uma lista de nomes e endereços IP dos servidores de DNS autoritativos para todos os Top-Level Domains (TLDs). Torna-se necessário incluir um ficheiro relativo aos servidores de nomes de raiz de forma a assegurar que, dada uma qualquer *query*, o servidor seja pelo menos capaz de facultar os nomes e endereços dos servidores autoritativos para o TLD em que o domínio pesquisado se encontra [1]. Dado que o ficheiro `named.root` existe obrigatoriamente em todas as três máquinas e a sua construção é semelhante para todos os casos, o respectivo conteúdo será apenas explicitado nesta secção.

```
;       last update:    Jun 8, 2011
;       related version of root zone:   2011060800
;
; formerly NS.INTERNIC.NET
;
.                       3600000  IN  NS    A.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET.     3600000      A     198.41.0.4
A.ROOT-SERVERS.NET.     3600000      AAAA  2001:503:BA3E::2:30
;
; FORMERLY NS1.ISI.EDU
;
.                       3600000      NS    B.ROOT-SERVERS.NET.
B.ROOT-SERVERS.NET.     3600000      A     192.228.79.201
;
; FORMERLY C.PSI.NET
;
.                       3600000      NS    C.ROOT-SERVERS.NET.
C.ROOT-SERVERS.NET.     3600000      A     192.33.4.12
;
; FORMERLY TERP.UMD.EDU
;
.                       3600000      NS    D.ROOT-SERVERS.NET.
D.ROOT-SERVERS.NET.     3600000      A     128.8.10.90
D.ROOT-SERVERS.NET.  3600000      AAAA  2001:500:2D::D
;
; FORMERLY NS.NASA.GOV
;
.                       3600000      NS    E.ROOT-SERVERS.NET.
E.ROOT-SERVERS.NET.     3600000      A     192.203.230.10
```

```
;
; FORMERLY NS.ISC.ORG
;
.                       3600000      NS    F.ROOT-SERVERS.NET.
F.ROOT-SERVERS.NET.     3600000      A     192.5.5.241
F.ROOT-SERVERS.NET.     3600000      AAAA  2001:500:2F::F
;
; FORMERLY NS.NIC.DDN.MIL
;
.                       3600000      NS    G.ROOT-SERVERS.NET.
G.ROOT-SERVERS.NET.     3600000      A     192.112.36.4
;
; FORMERLY AOS.ARL.ARMY.MIL
;
.                       3600000      NS    H.ROOT-SERVERS.NET.
H.ROOT-SERVERS.NET.     3600000      A     128.63.2.53
H.ROOT-SERVERS.NET.     3600000      AAAA  2001:500:1::803F:235
;
; FORMERLY NIC.NORDU.NET
;
.                       3600000      NS    I.ROOT-SERVERS.NET.
I.ROOT-SERVERS.NET.     3600000      A     192.36.148.17
I.ROOT-SERVERS.NET.     3600000      AAAA  2001:7FE::53
;
; OPERATED BY VERISIGN, INC.
;
.                       3600000      NS    J.ROOT-SERVERS.NET.
J.ROOT-SERVERS.NET.     3600000      A     192.58.128.30
J.ROOT-SERVERS.NET.     3600000      AAAA  2001:503:C27::2:30
;
; OPERATED BY RIPE NCC
;
.                       3600000      NS    K.ROOT-SERVERS.NET.
K.ROOT-SERVERS.NET.     3600000      A     193.0.14.129
K.ROOT-SERVERS.NET.     3600000      AAAA  2001:7FD::1
;
; OPERATED BY ICANN
;
.                       3600000      NS    L.ROOT-SERVERS.NET.
L.ROOT-SERVERS.NET.     3600000      A     199.7.83.42
L.ROOT-SERVERS.NET.     3600000      AAAA  2001:500:3::42
;
; OPERATED BY WIDE
```

```
;
.                       3600000      NS    M.ROOT-SERVERS.NET.
M.ROOT-SERVERS.NET.     3600000      A     202.12.27.33
M.ROOT-SERVERS.NET.     3600000      AAAA  2001:DC3::35
;
; End of File
```

2.1.2 Ficheiro de Configuração

O Ficheiro de Configuração `named.conf` controla o comportamento e a funcionalidade do `BIND` e é, neste caso particular, composto por três cláusulas distintas, que definem as zonas específicas que o servidor de nomes em questão suporta.

```
options {
directory "/home/tav/Desktop/master";
};

zone "." {

type hint;
file "/home/tav/Desktop/named.root";

};

zone "miebiom-g4.gcom.di.uminho.pt."
{

type master;
file "/home/tav/Desktop/master/zone1.com";
allow-transfer { 172.19.111.134; 127.0.0.1; };

};

zone "2.168.192.in-addr.arpa."
{

type slave;
masters { 172.19.111.134; };
file "/home/tav/Desktop/master/PRT_fileZone";
allow-transfer { 127.0.0.1; };

};
```

2.1.3 Ficheiro de Zona

O Ficheiro de Zona `zone1.com`, por sua vez, descreve autoritativamente uma zona de DNS e contém o mapeamento entre os *domain names* e os respectivos endereços de IP, entre outros recursos, organizados sob a forma de representações textuais de *Resource Records (RR)* [1]. Os parâmetros temporais neste declarados devem ser razoáveis, pelo que foram optimizados de forma a potenciar as condições de teste, como seja através da diminuição dos tempos de *Refresh* e *Retry* para 1 hora e 300 segundos, respectivamente.

De forma a garantir o funcionamento do servidor de DNS em caso de falha do servidor primário `ns1`, declarou-se um segundo servidor (secundário) `ns2`. Ambos foram implementados enquanto instâncias do tipo `NS (Name Server Record)`, tendo ainda sido incluídos registos do tipo `MX (Mail Exchange Record)` — associados a diferentes prioridades — e `A (Address Record)`.

```
$ORIGIN miebiom-g4.gcom.di.uminho.pt.     ;
$TTL 1h                  ;
miebiom-g4.gcom.di.uminho.pt. IN SOA ns1.miebiom-g4.gcom.di.uminho.pt.  username.miebiom-g
4.gcom.di.uminho.pt. ( 2012112902 ;
              1h         ;
              300        ;
              4w         ;
              1h         ;

        )

miebiom-g4.gcom.di.uminho.pt. NS    ns1.miebiom-g4.gcom.di.uminho.pt.  ;
miebiom-g4.gcom.di.uminho.pt. NS    ns2.miebiom-g4.gcom.di.uminho.pt. ;

miebiom-g4.gcom.di.uminho.pt. MX    10 mail1.miebiom-g4.gcom.di.uminho.pt. ;
@                             MX    20 mail2.miebiom-g4.gcom.di.uminho.pt. ;

ns1     A     172.19.111.254 ;
ns2     A     192.19.111.134 ;

mail1 A 192.168.1.1
mail2 A 192.168.2.1
```

2.2 Máquina 2

A Máquina 2 foi configurada como *master* do domínio reverso `2.168.192.in-addr.arpa.` e *slave* do domínio `miebiom-g4.gcom.di.uminho.pt`, gerido pela Máquina 1. Nesta fase, é, então, implementada uma das componentes fundamentais do processo de resolução: o mapeamento inverso; sendo que, ao domínio padrão `in-addr.arpa`, é acrescentado o endereço IP da máquina no sentido inverso — dado que, considerando a árvore do domínio, o nome é lido no sentido das folhas para a raiz.

2.2.1 Ficheiro de Configuração

O Ficheiro de Configuração `named.conf` apresenta, de forma semelhante, três cláusulas, relativas à zona `root`, à zona `miebiom-g4-gcom.di.uminho.pt.` e à zona `2.168.192.in-addr.arpa.`, para as quais a Máquina 2 é *slave* e *master*, respectivamente.

```
options {
directory "/Users/martapaesmoreira/Desktop/slave";
};

zone "." {
type hint;
file "named.root";
};

zone "miebiom-g4.gcom.di.uminho.pt" {
type slave;
masters { 172.19.101.45; };
file "/Users/martapaesmoreira/Desktop/slave/master_zonefile";
allow-transfer {127.0.0.1;};
};

zone "2.168.192.in-addr.arpa." {
type master;
file "/Users/martapaesmoreira/Desktop/slave/zone2.com";
allow-transfer { 172.19.101.45; 172.19.111.63; 127.0.0.1; };
};
```

2.2.2 Ficheiro de Zona

O Ficheiro de Zona `zone2.com` caracteriza autoritativamente a zona de mapeamento inverso, sendo que os tipos de *RR* utilizados são apenas o `NS` e o `PTR (Pointer Record)` — um apontador responsável por fazer o mapeamento entre um endereço IP e um *hostname*.

```
$TTL 1h
$ORIGIN 2.168.192.in-addr.arpa.
@ IN SOA ns1.miebiom-g4.gcom.di.uminho.pt. username.miebiom-g4.gcom.di.uminho.pt.
( 2012121400 ;
1h ;
300 ;
4w ;
1h ) ;
    IN NS ns1.miebiom-g4.gcom.di.uminho.pt. ;
    IN NS ns2.miebiom-g4.gcom.di.uminho.pt. ;
132 IN PTR ns1.miebiom-g4.gcom.di.uminho.pt. ;
100 IN PTR ns2.miebiom-g4.gcom.di.uminho.pt. ;
```

2.3 Máquina 3

A Máquina 3, por sua vez, foi configurada exclusiva e propositadamente como *slave* do domínio reverso, de forma a veicular um elemento diferenciador no que concerne as possibilidades de *queries* formuladas.

2.3.1 Ficheiro de Configuração

No seguimento do raciocínio anterior, o Ficheiro de Configuração `named.conf` encerra apenas duas cláusulas distintas: uma respeitante à zona de raiz e uma outra à zona `2.168.192.in-addr.arpa.`.

```
options {
directory "/Users/hugomes/Desktop/slave";
};

zone "." {

type hint;
file "named.root";

};

zone "2.168.192.in-addr.arpa."
{

type slave;
    masters { 172.19.111.134; };
file "PRT_fileZone.txt";
allow-transfer { 127.0.0.1; };

};
```

2.3.2 Ficheiro de Zona

Inicialmente, não existirá na Máquina 3 qualquer Ficheiro de Zona, embora, aquando da inicialização do servidor, se processe uma transferência de zona que armazenará na directoria indicada no `named.conf` — mais precisamente, num ficheiro de extensão `.txt` de nome `PRT_fileZone` — a informação respeitante à zona `2.168.192.in-addr.arpa.`, embora não necessariamente numa notação semelhante. Escusado será dizer que uma situação análoga ocorre aquando da inicialização dos servidores respeitantes às Máquinas 1 e 2, já que também estas estão declaradas, nos respectivos ficheiros `named.conf`, como *slaves* de um dado domínio.

3 Queries

Numa primeira fase, testou-se a conectividade entre as diferentes máquinas, através do comando `ping`, de forma a assegurar que o sistema se encontrava interconectado e que seria, consequentemente, possível efectuar as *queries* pretendidas. De seguida, emitiu-se uma sequência de comandos em cada uma das máquinas no sentido de inicializar cada um dos servidores (1.), verificar que o processo pretendido se encontra em execução (2.) — e, sempre que necessário, matar o processo (3.) — e resgatar informação acerca dos registos de DNS (4.).

1. `sudo named -c /Users/martapaesmoreira/Desktop/slave/named.conf`
2. `ps ax | grep named`

   ```
   1548   ??  Ss     0:00.02 named -c /Users/martapaesmoreira/Desktop/slave/named.conf
   1560 s000  R+     0:00.00 grep named
   ```

3. `sudo kill -9 1548`
4. `nslookup`

Destaque-se que o primeiro indicador dos de que o sistema se encontrava em pleno funcionamento era o surgimento do ficheiro de zona do *master* respectivo na directoria indicada para o efeito no ficheiro de configuração `named.conf`.

3.1 Queries efectuadas pela Máquina 1

Enquanto *master* do domínio `miebiom-g4.gcom.di.uminho.pt.` e *slave* do domínio `2.168.192.in-addr.arpa.`, esperar-se-ia que as *queries* dirigidas ao próprio servidor tanto acerca do domínio directo como do domínio reverso apresentassem uma resposta autoritativa, na medida em que esta detém os ficheiros para ambas as zonas — um original e outro que é uma cópia daquele presente na directoria do *master*. Relativamente às *queries* efectuadas à Máquina 2 acerca do domínio reverso, apenas havia que ter em conta o tipo de registos que a compõem — que impossibilitam, por exemplo, *queries* do tipo `MX` e `A`.

1. Query do tipo `NS` ao servidor em execução na Máquina 1, acerca do domínio `miebiom-g4.gcom.di.uminho.pt.`.

   ```
   server localhost
   Default server: localhost
   Address: 127.0.0.1#53
   set ty=ns
   miebiom-g4.gcom.di.uminho.pt.
   Server: localhost
   Address: 127.0.0.1#53

   miebiom-g4.gcom.di.uminho.pt nameserver = ns1.miebiom-g4.gcom.di.uminho.pt.
   miebiom-g4.gcom.di.uminho.pt nameserver = ns2.miebiom-g4.gcom.di.uminho.pt.
   ```

2. Query do tipo `NS` ao servidor em execução na Máquina 1, acerca do domínio `2.168.192.in-addr.arpa.`.

```
2.168.192.in-addr.arpa.
Server: localhost
Address: 127.0.0.1#53

2.168.192.in-addr.arpa nameserver = ns1.miebiom-g4.gcom.di.uminho.pt.
2.168.192.in-addr.arpa nameserver = ns2.miebiom-g4.gcom.di.uminho.pt.
```

3. Query do tipo `MX` ao servidor em execução na Máquina 1, acerca do domínio `miebiom-g4.gcom.di.uminho.pt.`.

```
set ty=mx
miebiom-g4.gcom.di.uminho.pt
Server: localhost
Address: 127.0.0.1#53

miebiom-g4.gcom.di.uminho.pt mail exchanger = 20 mail2.miebiom-g4.gcom.di.uminho.pt.
miebiom-g4.gcom.di.uminho.pt mail exchanger = 10 mail1.miebiom-g4.gcom.di.uminho.pt.
```

4. Query do tipo `SOA` ao servidor em execução na Máquina 1, acerca do domínio `miebiom-g4.gcom.di.uminho.pt.`.

```
set ty=soa
miebiom-g4.gcom.di.uminho.pt.
Server: localhost
Address: 127.0.0.1#53

miebiom-g4.gcom.di.uminho.pt
origin = ns1.miebiom-g4.gcom.di.uminho.pt
mail addr = username.miebiom-g4.gcom.di.uminho.pt
serial = 2012112902
refresh = 3600
retry = 300
expire = 2419200
minimum = 3600
```

5. Query do tipo `SOA` ao servidor em execução na Máquina 1, acerca do domínio `2.168.192.in-addr.arpa.`.

```
2.168.192.in-addr.arpa.
Server: localhost
Address: 127.0.0.1#53

2.168.192.in-addr.arpa
origin = ns1.miebiom-g4.gcom.di.uminho.pt
```

```
mail addr = username.miebiom-g4.gcom.di.uminho.pt
serial = 2012121400
refresh = 3600
retry = 300
expire = 2419200
minimum = 3600
```

6. Query do tipo `PTR` ao servidor em execução na Máquina 1, acerca do endereço `192.168.2.100.`.

```
set ty=ptr
192.168.2.100
Server: localhost
Address: 127.0.0.1#53

100.2.168.192.in-addr.arpa name = ns2.miebiom-g4.gcom.di.uminho.pt.
```

7. Query do tipo `NS` ao servidor em execução na Máquina 2, acerca do domínio `miebiom-g4.gcom.di.uminho.pt.`.

```
server 172.19.111.134
Default server: 172.19.111.134
Address: 172.19.111.134#53
set ty=ns
miebiom-g4.gcom.di.uminho.pt.
Server: 172.19.111.134
Address: 172.19.111.134#53

miebiom-g4.gcom.di.uminho.pt nameserver = ns1.miebiom-g4.gcom.di.uminho.pt.
miebiom-g4.gcom.di.uminho.pt nameserver = ns2.miebiom-g4.gcom.di.uminho.pt.
```

8. Query do tipo `NS` ao servidor em execução na Máquina 2, acerca do domínio `2.168.192.in-addr.arpa.`.

```
2.168.192.in-addr.arpa.
Server: 172.19.111.134
Address: 172.19.111.134#53

2.168.192.in-addr.arpa nameserver = ns1.miebiom-g4.gcom.di.uminho.pt.
2.168.192.in-addr.arpa nameserver = ns2.miebiom-g4.gcom.di.uminho.pt.
```

9. Query do tipo `MX` ao servidor em execução na Máquina 2, acerca do domínio `2.168.192.in-addr.arpa.`.

```
set type=mx
miebiom-g4.gcom.di.uminho.pt.
Server: 172.19.111.134
Address: 172.19.111.134#53
```

```
miebiom-g4.gcom.di.uminho.pt mail exchanger = 20 mail2.miebiom-g4.gcom.di.uminho.pt.
miebiom-g4.gcom.di.uminho.pt mail exchanger = 10 mail1.miebiom-g4.gcom.di.uminho.pt.
```

10. Query do tipo `MX` ao servidor em execução na Máquina 2, acerca do domínio `2.168.192.in-addr.arpa.`.

```
2.168.192.in-addr.arpa.
Server: 172.19.111.134
Address: 172.19.111.134#53

*** Can't find 2.168.192.in-addr.arpa.: No answer
```

11. Query do tipo `SOA` ao servidor em execução na Máquina 2, acerca do domínio `miebiom-g4.gcom.di.uminho.pt`.

```
set ty=soa
miebiom-g4.gcom.di.uminho.pt
Server: 172.19.111.134
Address: 172.19.111.134#53

miebiom-g4.gcom.di.uminho.pt
origin = ns1.miebiom-g4.gcom.di.uminho.pt
mail addr = username.miebiom-g4.gcom.di.uminho.pt
serial = 2012112902
refresh = 3600
retry = 300
expire = 2419200
minimum = 3600
```

12. Query do tipo `SOA` ao servidor em execução na Máquina 2, acerca do domínio `miebiom-g4.gcom.di.uminho.pt`.

```
2.168.192.in-addr.arpa.
Server: 172.19.111.134
Address: 172.19.111.134#53

2.168.192.in-addr.arpa
origin = ns1.miebiom-g4.gcom.di.uminho.pt
mail addr = username.miebiom-g4.gcom.di.uminho.pt
serial = 2012121400
refresh = 3600
retry = 300
expire = 2419200
minimum = 3600
```

13. Query do tipo `PTR` ao servidor em execução na Máquina 2, acerca do domínio `miebiom-g4.gcom.di.uminho.pt`.

```
set ty=ptr
miebiom-g4.gcom.di.uminho.pt.
Server: 172.19.111.134
Address: 172.19.111.134#53

*** Can't find miebiom-g4.gcom.di.uminho.pt.: No answer
```

3.2 Queries efectuadas pela Máquina 2

Neste caso, a máquina em questão apresenta-se como *master* do domínio `2.168.192.in-addr.arpa.` e *slave* do domínio `miebiom-g4.gcom.di.uminho.pt.`, pelo que se previa um comportamento semelhante ao do caso explicitado anteriormente, no que concerne as *queries* dirigidas ao próprio servidor. Note-se que as *queries* dirigidas ao servidor da Máquina 3 acerca do domínio `miebiom-g4.gcom.di.uminho.pt` esperam uma resposta não autoritativa, na medida em que este servidor não está definido como *slave* da zona correspondente, pelo que não terá 'conhecimento de causa' para dar respostas acerca deste domínio sem consultar previamente um servidor que o tenha.

1. Query do tipo `PTR` ao servidor em execução na Máquina 2, acerca do endereço `192.168.2.100`:

```
server localhost
Default server: localhost
Address: ::1#53
Default server: localhost
Address: fe80::1#53
Default server: localhost
Address: 127.0.0.1#53
set type=ptr
192.168.2.100
Server: localhost
Address: 127.0.0.1#53

100.2.168.192.in-addr.arpa name = ns2.miebiom-g4.gcom.di.uminho.pt.
```

2. Query do tipo `NS` ao servidor em execução na Máquina 2 acerca do domínio `2.168.192.in-addr.arpa.`.

```
set type=ns
2.168.192.in-addr.arpa.
Server: localhost
Address: 127.0.0.1#53

2.168.192.in-addr.arpa nameserver = ns1.miebiom-g4.gcom.di.uminho.pt.
2.168.192.in-addr.arpa nameserver = ns2.miebiom-g4.gcom.di.uminho.pt.
```

3. Query do tipo `SOA` ao servidor em execução na Máquina 2, acerca do domínio `miebiom-g4.gcom.di.uminho.pt`:

```
set type=soa
miebiom-g4.gcom.di.uminho.pt
Server: localhost
Address: 127.0.0.1#53

miebiom-g4.gcom.di.uminho.pt
origin = ns1.miebiom-g4.gcom.di.uminho.pt
mail addr = username.miebiom-g4.gcom.di.uminho.pt
serial = 2012112902
refresh = 3600
retry = 300
expire = 2419200
minimum = 3600
```

4. Query do tipo `MX` ao servidor em execução na Máquina 2, acerca do domínio `miebiom-g4.gcom.di.uminho.pt`:

```
set type=mx
miebiom-g4.gcom.di.uminho.pt
Server: localhost
Address: 127.0.0.1#53

miebiom-g4.gcom.di.uminho.pt mail exchanger = 20 mail2.miebiom-g4.gcom.di.uminho.pt.
miebiom-g4.gcom.di.uminho.pt mail exchanger = 10 mail1.miebiom-g4.gcom.di.uminho.pt.
```

5. Query do tipo `NS` ao servidor em execução na Máquina 3, acerca do domínio `miebiom-g4.gcom.di.uminho.pt`:

```
server 172.19.111.63
Default server: 172.19.111.63
Address: 172.19.111.63#53
set type=ns
miebiom-g4.gcom.di.uminho.pt
Server: 172.19.111.63
Address: 172.19.111.63#53

miebiom-g4.gcom.di.uminho.pt nameserver = ns2.miebiom-g4.gcom.di.uminho.pt.
miebiom-g4.gcom.di.uminho.pt nameserver = ns1.miebiom-g4.gcom.di.uminho.pt.
```

6. Query do tipo `SOA` ao servidor em execução na Máquina 3, acerca do domínio `miebiom-g4.gcom.di.uminho.pt`:

```
set type=soa
miebiom-g4.gcom.di.uminho.pt
Server: 172.19.111.63
Address: 172.19.111.63#53

miebiom-g4.gcom.di.uminho.pt
origin = ns1.miebiom-g4.gcom.di.uminho.pt
mail addr = username.miebiom-g4.gcom.di.uminho.pt
serial = 2012112902
refresh = 3600
retry = 300
expire = 2419200
minimum = 3600
```

3.3 Queries efectuadas pela Máquina 3

O servidor alojado na Máquina 3 apresenta-se apenas como *slave* da zona `2.168.192.in-addr.arpa.`, sendo que apenas se encontra habilitado a responder autoritativamente a *queries* acerca desta, não podendo, porém, responder a *queries* sobre si próprio, na medida em que não existe nenhuma zona que lhe esteja originalmente associada.

1. Query do tipo `NS` ao servidor em execução na Máquina 2, acerca do domínio `miebiom-g4.gcom.di.uminho.pt`:

   ```
   server 172.19.111.134
   Default server: 172.19.111.134
   Address: 172.19.111.134#53

   set type=ns
   miebiom-g4.gcom.di.uminho.pt
   Server: 172.19.111.134
   Address: 172.19.111.134#53

   miebiom-g4.gcom.di.uminho.pt nameserver = ns2.miebiom-g4.gcom.di.uminho.pt.
   miebiom-g4.gcom.di.uminho.pt nameserver = ns1.miebiom-g4.gcom.di.uminho.pt.
   ```

2. Query do tipo `SOA` ao servidor em execução na Máquina 2, acerca do domínio `miebiom-g4.gcom.di.uminho.pt`:

   ```
   set type=soa
   miebiom-g4.gcom.di.uminho.pt
   Server: 172.19.111.134
   Address: 172.19.111.134#53

   miebiom-g4.gcom.di.uminho.pt
   ```

```
origin = ns1.miebiom-g4.gcom.di.uminho.pt
mail addr = username.miebiom-g4.gcom.di.uminho.pt
serial = 2012112902
refresh = 3600
retry = 300
expire = 2419200
minimum = 3600
```

4 Conclusão e Recomendações Futuras

Implementou-se com sucesso um serviço de DNS que compreende três servidores distintos — nas Máquina 1, 2 e 3 — dotados de conectividade e, por conseguinte, capazes de dirigir *queries* entre si. Os ficheiros básicos que asseguram o funcionamento do sistema (cuja implementação se revela simples e pouco complexa) e devem existir em cada um dos servidores são o Ficheiros de Raiz e o Ficheiro de Configuração, sendo que os *masters* de cada um dos domínios devem ainda possuir um Ficheiro de Zona, que será posteriormente transferido aos respectivos *slaves*. As principais dificuldades encontradas ao nível da construção dos ficheiros prenderam-se com erros de sintaxe, detectados e corrigidos recorrendo à funcionalidade de `debug` e à consulta dos ficheiros de `system.log`.

Relativamente às respostas a *queries* dadas por cada um dos servidores, e considerando a sua tipologia específica, é possível inferir que um servidor não é necessariamente *master* ou *slave*, podendo apresentar uma tipologia distinta para diferentes domínios, conforme se comprova através dos servidores implementados nas Máquinas 1 e 2. No seguimento desta inferência, conclui-se ainda que tanto o *master* como o *slave* podem dar repostas autoritativas acerca do respectivo domínio de referência, dado que, no primeiro caso, o servidor possui originalmente a informação no Ficheiro de Zona e, no segundo, dá-se uma transferência de zona que veicula ao servidor uma cópia da informação contida no Ficheiro de Zona do *master*. No caso específico de *queries* dirigidas ao servidor do domínio reverso — alojado na Máquina 2 —, apenas se logrou, conforme esperado, respostas àquelas de tipo `NS`, `PTR` e `SOA`, o mesmo acontecendo para o domínio directo, que apenas responde a `queries` dos tipos `NS`, `MX`, `A` e `SOA`. Por sua vez, o terceiro servidor implementado cumpre o propósito inicialmente definido, repondendo apenas autoritativamente a *queries* acerca do servidor do domínio reverso, configurado como seu *master*.

No âmbito da segurança, pode concluir-se que os principais objectivos das tentativas de ataque a um servidor de nomes se prendem, por um lado, com o endereçamento do cliente através de um percurso distinto daquele pretendido originalmente — *client misdirection* —, sendo é possível efectuar esta redirecção empregando as técnicas de `DNS Spoofing`, `DNS ID Hacking` e `DNS Cache Poisoning`; e, por outro lado, com a provocação da falha dos servidores, através da exploração de *bugs* no *software*. No sentido de evitar os efeitos destes ataques num contexto prático, devem considerar-se medidas de optimização de segurança, como sejam a actualização frequente da versão do `BIND` utilizada e a restrição dos *hosts* que podem emitir *queries* ao servidor. Pese embora a versão do `BIND` utilizada — 9.8.4-P1 — seja bastante recente, crê-se que existam lacunas significativas ao nível da segurança do serviço implementado, que poderiam ser futuramente colmatadas através, por exemplo, da utilização de *Access Control Lists (ACLs)* e da implementação dos servidores em sub-redes distintas.

Referências

[1] P. Albitz C. Liu. Dns and bind. O'Reilly, 1998. www.cse.unl.edu/~ylu/csce855/.../OReilly-DNS-and-BIND.pdf.

[2] B. King A. Householder. Securing an internet name server. CERT Coordination Center, 2002. www.cert.org/archive/pdf/dns.pdf.

[3] Bind 9 administrator reference manual. Internet Systems Consortium, 2012. http://theory.stanford.edu/~csilvers/cs359/.