Enterprise Linux 實戰講座

Domain Name System　網域名稱伺服器（二）

前言

上一期文章，筆者並未提到反解的部份及 Slave DNS。其實建置 DNS 不一定需要設定反解 Zone，而且即使你設定了反解的 Zone，也不見得有用。為什麼這樣說呢？這期文章將探討反解（Reverse Lookup）的原理，讀者就可以明白其中道理，接著介紹 DNS 主要設定檔 named.conf 中的細項參數及如何實作 Slave DNS。

DNS 反解（Reverse Lookup）原理

DNS 除了提供查詢網路上「主機名稱」所對應的 IP Address 正解（Forward Lookup）的功能外，也具備將 IP Address 反推「主機名稱」的反解服務（Reverse Lookup）。

圖 1 為 DNS 反解的流程圖，假設想要知道 IP Address 211.21.98.10 的名稱為何？反解（使用 PTR Record）查詢過程跟正解流程雷同，DNS 會先檢查此 IP 是否是本身所負責的網段？快取暫存區中是否有相關之紀錄？若皆不是則將此需求傳給 root server， root server 將需求往下丟給 arpa server（讀者可以把 arpa server 想成就是一台 DNS），然後再往下給 in-addr server，再到負責所有 IP Address 211 開頭的伺服器，依此步驟最後到負責 211.21.98 網段的伺服器，由其反解 Zone 得知 211.21.98.10 所對應的主機名稱為 www.test.com.tw。

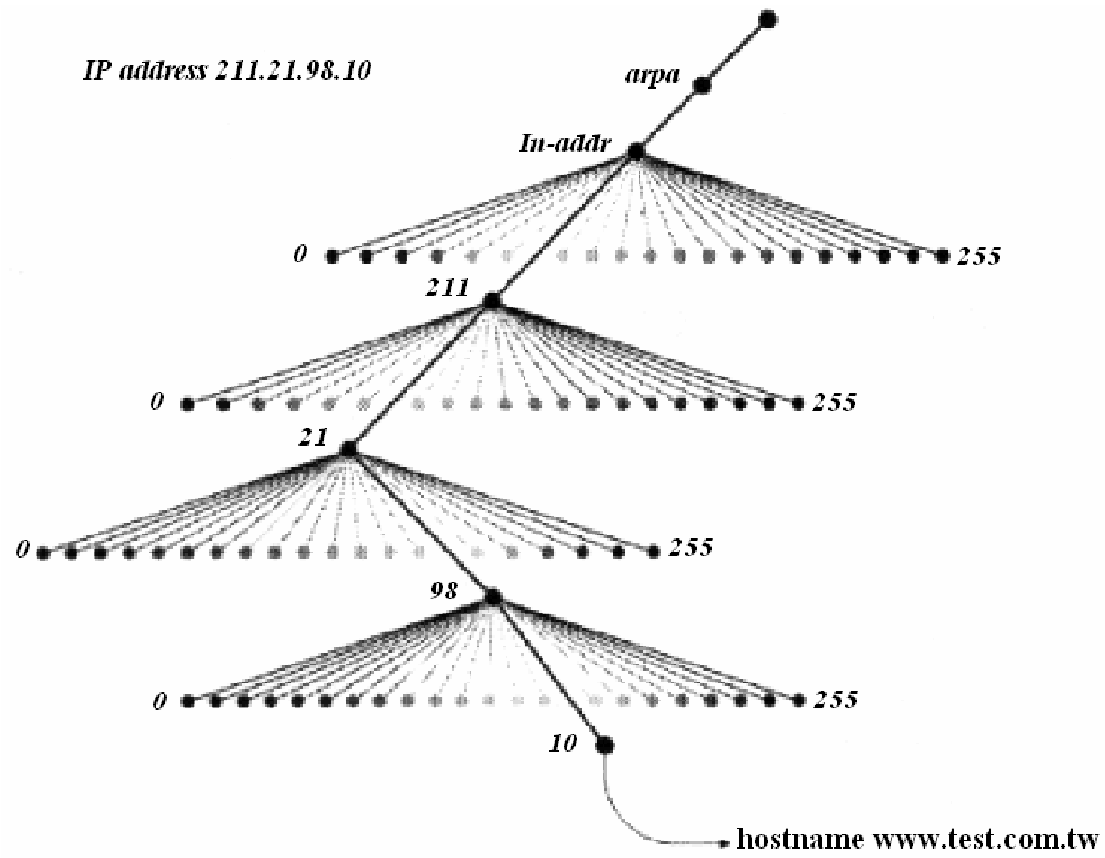

圖 1：DNS 反解（Reverse Lookup）原理圖

讀者可能會好奇，什麼時候會用到反解，電腦會作反解的行為主要有下列兩個目的：

1、讓使用者或管理者容易了解連線情形；如在畫面或 Log 中顯示主機名稱（FQDN）會比顯示 IP Address 讓人更容易了解連線對象是那個單位

2、安全考量；如某些 server 會以連線對象的 IP Address 查詢主機名稱(FQDN)，再檢查主機名稱（FQDN）對應的 IP Address 是否一致。

根據 TWNIC 的調查，國內 IP 的反解比率約四成，而中國大陸不到 5%，造就國內高反解比率(與國際相較)主要因為 TANET 早期的推動及近來 ISP 的處理方式。

國內的系統較不嚴謹，通常不會檢查正反解的一致性，但國外蠻高比例的系統都會進行這個部分的確認；由來源 IP 查反解名稱，依結果再查正解，並檢驗其結果是否一致。例如，有部分的 Mail Server 也會使用正反解確認的機制來減少 SPAM 的問題。

很多人會認為網路上的服務正解的需求高於反解的需求，其實不盡然。根據調查 DNS Query 正解部份約佔 40%，反解佔 60%。原因是多數的服務皆會進行 IP 來源的反查所致（Ex：WWW、MAIL、Firewall …）。

還記得在上一期文章中，筆者曾提到「**為什麼執行 telnet 或 ftp IP Address，會經過很久才出現登入畫面？**」。原因通常是 /etc/resolv.conf 或 /etc/hosts 設定錯誤或 DNS 異常所造成，因為許多網路服務會檢查反解是否設立，通常不成立也一樣可以建立連線，但是會因為等待 timeout，而造成很久才出現登入畫面，由此可發現反解的運作無所不在。

實戰演練一：建置 Master DNS（含反解區域）

目的：上期文章實戰演練三中已實作 Master DNS，但不含反解區域。此演練假設此 DNS 亦提供反解功能，所負責的網段為 61.219.23.0/24。而所負責的網域（Domain）跟上期文章一樣為「blue-linux.com」。此伺服器的主機名稱為「dns.blue-linux.com」，IP Address 為「61.219.23.88」。

實作環境:RHEL 3

註：RHEL 4 若關掉 chroot 機制，則作步驟亦同 RHEL 3
#vi /etc/sysconfig/named
在 ROOTDIR=/var/named/chroot 前加上#，然後#service named restart 即可。
若不關閉此功能，則所有設定檔皆需放在/var/named/chroot 目錄下，就是把 /var/named/chroot 目錄想成/ 目錄。所以在 RHEL 3 上要修改/etc/named.conf；在 RHEL 4 上就得修改/var/named/chroot/etc/named.conf。

步驟一：修改/etc/named.conf 加入負責網段 61.219.23.0/24 之設定

```
[root@dns ~]#vi /etc/named.conf
// generated by named-bootconf.pl

options {
        directory "/var/named";
         /*
         * If there is a firewall between you and nameservers you want
```

```
            * to talk to，  you might need to uncomment the query-source
            * directive below.   Previous versions of BIND always asked
            * questions using port 53，but BIND 8.1 uses an unprivileged
            * port by default.
            */
           // query-source address * port 53;
};

//
// a caching only nameserver config
//
controls {
           inet 127.0.0.1 allow { localhost; } keys { rndckey; };
};
zone "." IN {
           type hint;
           file "named.ca";
};

zone "localhost" IN {
           type master;
           file "localhost.zone";
           allow-update { none; };
};

zone "0.0.127.in-addr.arpa" IN {
           type master;
           file "named.local";
           allow-update { none; };
};
#下面這段文字是在前一期文章演練三所加入
zone "blue-linux.com" {
           type master;
           file "db.blue-linux";
};

/* 此次演練再加入下面這段文字代表負責的網段為 61.219.23.0/24，反解 Zone File 為 db.61.219.23 */
```

```
zone "23.219.61.in-addr.arpa" {
        type master;
        file "db.61.219.23";
};
include "/etc/rndc.key";
```

步驟二：編寫反解 Zone File

在/var/named 目錄下除了之前撰寫的 db.blue-linux 外，再編寫名為的 db.61.219.23 的反解 Zone File。兩個檔案的內容如下：

```
[root@dns /var/named]#vi db.blue-linux
$TTL 86400
@ IN    SOA dns.blue-linux.com. alex.blue-linux.com. (
                        2004112001      ; serial number
                        10800           ;refresh
                        3600            ;retry
                        604800          ;expire
                        0 )             ;TTL for negative answer

blue-linux.com.    IN    NS    dns.blue-linux.com.
;加上下列的 Resource Record
dns.blue-linux.com.     IN      A               61.219.23.88
dns2.blue-linux.com.    IN      A               61.219.23.89
mail1.blue-linux.com.   IN      A               61.219.23.90
mail2.blue-linux.com.   IN      A               61.219.23.91
ftp                     IN      CNAME           dns
;如果第一個欄位沒有以 . 結束的話，會自動加上這個網域名稱；所以 ftp 相當於 ftp.blue-linux.com.
blue-linux.com.         IN      MX      10      mail1
blue-linux.com.         IN      MX      20      mail2
;當有信件要寄送至 blue-linux.com.(即 email 格式為 user@blue-linux.com)時，會將信件送至優先權較高（數值較小）的 mail1；當 mail1 當掉時，信件才會送至 mail2。所以通常 mail2 為備援的 Mail Server

station1                IN      A               61.219.23.101
station2                IN      A               61.219.23. 102
station3                IN      A               61.219.23.103
station4                IN      A               61.219.23.104
```

```
station5                        IN          A               61.219.23.105
;上面 5 行具有規則性，可利用$GENERATE 變數將定檔簡化為下面的格式
;$GENERATE 1-5    station$          A               61.219.23.10$
;station $ 即表示station1-5，將會自動展開成 5 行的 A Record
;Class 欄位不可寫入 IN
```

```
[root@dns /var/named]# vi db.61.219.23
$TTL 86400
@ IN    SOA dns.blue-linux.com. root.dns.blue-linux.com. (
                        2004112001          ; serial number
                        10800               ;refresh
                        3600                ;retry
                        604800              ;expire
                        0 )                 ;TTL for negative answer
                                   IN          NS    dns.blue-linux.com.
88.23.219.61.in-addr.arpa.          IN          PTR     dns.blue-linux.com.
89.23.219.61.in-addr.arpa.          IN          PTR     dns2.blue-linux.com.
90.23.219.61.in-addr.arpa.          IN          PTR     mail1.blue-linux.com.
91.23.219.61.in-addr.arpa.          IN          PTR     mail2.blue-linux.com.

101                                  IN          PTR    station1.blue-linux.com.
102                                  IN          PTR    station2.blue-linux.com.
103                                  IN          PTR    station3.blue-linux.com.
104                                  IN          PTR    station4.blue-linux.com.
105                                  IN          PTR    station5.blue-linux.com.
;如果第一個欄位沒有以 . 結束的話，會自動加上.23.219.61.in-addr.arpa.
;上面 5 行具有規則性，可利用$GENERATE 變數將定檔簡化為下面的格式
;$GENERATE      1-5  10$                         PTR     station$.example.com.
; $ 即表 1-5，上行將會自動展開成 5 行的 PTR Record
; Class 欄位不可寫入 IN
```

步驟三：重新讀取設定檔及測試反解結果

```
[root@dns /var/named]# service named reload
重新載入 named:                                                   [  確定  ]
[root@dns /var/named]# host 61.219.23.88
88.23.219.61.in-addr.arpa domain name pointer dns.blue-linux.com.
```

```
[root@dns /var/named]# host 61.219.23.89
89.23.219.61.in-addr.arpa domain name pointer dns2.blue-linux.com.
[root@dns /var/named]# host 61.219.23.101
101.23.219.61.in-addr.arpa domain name pointer station1.example.com.
[root@dns /var/named]# host 61.219.23.102
102.23.219.61.in-addr.arpa domain name pointer station2.example.com.
```

雖然在伺服器本身測試反解結果正確，此時讀者一定會有疑問？由圖一來看，不是應該向上一層反解 DNS 註冊，請它把 61.219.23.0/24 授權給你管理？沒錯，所以剛剛實作的反解的部份，若上一層 DNS 未授權 61.219.23.0/24 給此台 DNS，則網路上的 DNS 是不會來詢問有關 61.219.23.0/24 反解的資訊。但在現今 IP 不敷使用的情況下，要申請到一個 Class C 談何容易！難道沒有一個 Class C，就無法設定反解嗎？

在台灣未滿一個 Class C，必須麻煩 ISP 代為設定反解紀錄，例如 HiNet 的用戶可到下列網址 http://hidomain.hinet.net/hidns.html 直接填寫相關設定，此網址並有提供申請反請範例，可參考圖 2（http://hidomain.hinet.net/hidns.html）。

圖 2：用戶申請由 HiNet 建立領域反解範例

其實反解授權還是可以小於 Class C，只是台灣的 ISP 不願如此幫客戶設定。假設欲將 61.219.23.0/24 切割為 16 個 Subnet（即/28），分配給 16 家公司。

首先，在負責 61.219.23.0/24 網段 DNS 的反解 Zone File 將這個 Class C 切割為 16 個 Subnet，並將其授權（Delegation）給這 16 家公司的 DNS。以演練一為例，便是修改 db.61.219.23， 加入下列設定：

```
[root@dns /var/named]# vi db.61.219.23
$TTL 86400
@ IN    SOA dns.blue-linux.com. root.dns.blue-linux.com. (
                        2004112001      ; serial number
                        10800           ;refresh
                        3600            ;retry
                        604800          ;expire
                        0 )             ;TTL for negative answer
$ORIGIN     23.219.61.in-addr.arpa.; 預設附加字尾
；$ORIGIN 設定之後就可以使用@符號來代表所管理網域或網段，設定上此較清楚，也容易換名字。
                                IN      NS    dns.blue-linux.com.
; 將 16 個子網授權出去
;第一家公司
company1      IN      NS      company1_ns.xxx.com.tw.
;第二家公司
company2      IN      NS      company2_ns.yyy.com.tw.
;依此類推 16 家公司…

;以 $GENERATE 的方式，建立 CNAME ，將查詢轉往 16 個子網段上 DNS:
$GENERATE 0-15     $     CNAME          $.company1
$GENERATE 16-31    $     CNAME          $.company2
;依此類推 16 家公司

;當有人查詢 61.219.23.1 之反解時，
;會查到其 CNAME 至 1. company1.23.219.61.in-addr.arpa
;此時 company1 公司的 DNS 對等反解網段應定義為
;zone "company1. 23.219.61.in-addr.arpa";
```

接下來，必須在 company1. 23.219.61.in-addr.arpa (擁有 61.219.23.0~15 之單位) 的 DNS（就是 company1 的 DNS）上面須設立該反 Zone， Zone File 之內容再指出其反解（PTR）的結果為何。

```
[root@compnay1-dns ~]#vi /etc/named.conf
#加入下列文字
zone  "company1. 23.219.61.in-addr.arpa"  {
      type master;
      file  "db.0-15.23.219.61";
};
```

```
[root@compnay1-dns ~]#vi /var/named/ db.0-15.23.219.61
;file db.0-15.23.219.61
;SOA/NS Record 省略
$ORIGIN company1.23.219.61.in-addr.arpa.
1     IN     PTR     station1.company1.com.tw.
2     IN     PTR     station2.company1.com.tw.
```

可以看得出，小於 Class C 之反解設定較為複雜，反解設定由你取得 IP 的 ISP 負責，ISP 應提供一般反解的網頁設定。而多數用戶認為反解不重要，其實有些應用程式（如某些 mail server）會檢查正反解是否一致，若不一致可能會被拒絕連線，所以正確的設定反解 Record 確是不容輕忽。

named.conf 細項參數及 ACL（Access Control List）

接下來探討/etc/named.conf 這個 DNS 主要設定的細項參數及利用 ACL 讓 named.conf 設定更加清楚有效率。

■ Options 全域項目設定

options {

[directory path_name;]

Zone file 的預設存放位置（預設為/etc）

[named-xfer path_name;]

slave AXFR 存放的位置（預設為/etc）

[dump-file path_name;]

core dump 預設位置（預設為/etc）

[pid-file path_name;]

named 的 PID 存放位置

[auth-nxdomain yes_or_no;]

是否保留負面資料（預設為 no），即不正確資訊的狀況是否做快取（Cache）

[fake-iquery yes_or_no;]

作假DNS server的反解（預設為no）

[fetch-glue yes_or_no;]

不做任何的cache（預設為no）

[multiple-cnames yes_or_no;]

一個 FQDN 可否做IN CNAME 多次

[notify yes_or_no;]

Zone 變更通知（預設為yes）

[recursion yes_or_no;]

遞迴查詢，回應問的人去哪裏查（預設為yes）

[forward (only | first);]

Only代表只使用Forward first 則先使用Forward

[forwarders { [in_addr ; [in_addr ; ...]] };]

找不到使資料都往該 IP（另一台 DNS）送， 若此項有值則上一個項目預設為 first。

[check-names (master | slave | response) (warn | fail | ignore);]

檢查 FQDN 名稱不合法性於 type 為 (master | slave | 查詢要求)就(警告|失敗|忽略)

[allow-query { address_match_list };]

允許從那些 IP 查詢，可使用 ACL Name

[allow-transfer { address_match_list };]

允許從 IP AXFR（Zone Transfer），可使用 ACL Name

[listen-on [port ip_port] {address_match_list };]

DNS Listen port 為何，IP為何，不建議更改

[query-source [address(ip_addr|*)][port(ip_port|*)];]

查詢外部的 DNS(IP|*)時使用 ip_port，不建議更改

[max-transfer-time-in number;]

AXFR（Zone Transfer）的最大分鐘（預設為 120m）

[transfer-format(one-answer|many-answers);]

AXFR（Zone Transfer）時一次幾筆 RR（預設為 one-answer）

[transfers-in number;]

同時間最大的 AXFR（Zone Transfer）in 數目（預設為=10）

[transfers-out number;]

同時間最大的 AXFR（Zone Transfer）out 數目（預設為=10）

[transfers-per-ns number;]

每部 NS 同時間 AXFR（Zone Transfer）為 N 個

[version “version_string”;]

版本說明，隱藏版本有助於系統安全

[use-id-pool yes_or_no;]

每個查詢都保持一份 query ID（預設為 no），會增加系統負擔但能增加安全性

[blackhole { address_match_list };]

來自這些 IP 的查詢將不必處理

[lame-ttl number;]

不良的委任資料記錄要保留多久秒（0~18000 不留， 預設為 600）

[max-ncache-ttl number;]

負面資料的快取秒數（預設為 10800(3H) ， N<7D）

■Zone 轄區設定

Master Zone

```
zone “domain_name” {
type master; 類別為主要Master;表示權威主機
file path_name; Zone File的檔名
[ check-names ( warn | fail | ignore ); ]
[ allow-update { address_match_list }; ] 允許來自IP 的動態動新(使用
nsupdate 指令)
[ allow-query { address_match_list }; ]
[ allow-transfer { address_match_list }; ]
[ notify yes_or_no; ] 轄區資料變更是否同通知Slave主機(NS RR)
[ also-notify { ip_addr; [ ip_addr; ... ] };轄區資料變更時通知那些DNS主機( IP )
};
```

Slave Zone

```
zone “domain_name” {
type slave;
[ file path_name; ]
masters [ port ip_port ]{ ip_addr;[ip_addr;..]};
[ check-names ( warn | fail | ignore ); ]
[ allow-update { address_match_list }; ]
[ allow-query { address_match_list }; ]
[ allow-transfer { address_match_list }; ]
[ notify yes_or_no; ]
[ also-notify { ip_addr; [ ip_addr; ... ] };
};
```

Forward Zone

```
zone “domain_name” {
```

```
type forward;
[ forward ( only | first ); ]
[ forwarders { [ ip_addr ; [ ip_addr ; ... ] ] }; ]
[ check-names ( warn | fail | ignore ); ]
};
```

ACL 存取控制列表

主要在定義存取的列表，供其他"參數"所使用

```
acl acl_name {
IP; IP 位址 IP/[netmask]
DN; 網域名稱 *.blue-linux.com
path_name; 檔案名稱， 內存 ACL
CIDR; IP 段 61.219.23.0/24
None; 沒有任何 IP
Any; 任何 IP
Localhost; localhost ( 127.0.0.1)
Localnets; 網卡的IP/Netmask（即相連的網路）
```

例如:

acl Internal {192.168.0.0/24; 61.219.23.0/24;}

在別的“參數”再定義其行為

allow-transfer { Internal;};

allow-query { Internal;};

實戰演練二：建置 Slave DNS

在介紹完 named.conf 的細項設定後，就可以實作 Slave DNS，因為建置 Slave DNS 的最主要工作在於修改 named.conf。

Slave DNS 功能最主要為備份 Master DNS 的資料庫，並提供名稱解析的功能。它本身也有網域的 Zone File，不過它的 Zone File 是向 Master DNS 複製（Zone Transfer）而來的。所以在 Slave DNS 的設定中最主要有兩件事：1.設定 DNS Type 為 Slave。2.指定 Master DNS 的 IP。

實作環境：

作業系統：RHEL ES 3.0 版

Master DNS：dns.blue-linux.com（61.219.23.88）

Slave DNS：dns2.blue-linux.com（61.219.23.89）

【註：Master DNS 已在前幾個實戰演練中建置完成】

實作步驟：

步驟一：安裝相關套件

跟 DNS 相關套件如下：

```
bind-utils-9.2.2-21
bind-9.2.2-21
caching-nameserver-7.2-7
redhat-config-bind-2.0.0-14
```

■ bind-utils-9.2.2-21 內為 host、dig、nslookup 等 DNS 查詢必備工具及 DNS 所需的 library。
■ bind-9.2.2-21 內為 BIND 9.2.2 主要程式。
■ redhat-config-bind-2.0.0-14 為圖形化的 DNS 設定工具。
■ caching-nameserver-7.2-7 提供設定 Caching-Only DNS 所需的設定檔。

bind-utils-9.2.2-21，bind-9.2.2-21，caching-nameserver-7.2-7 這三個套件一定要安裝。讀者可利用「**rpm –ivh 套件檔案名稱**」指令進行安裝或是執行「**redhat-config-packages**」勾選「名稱伺服器」，安裝相關套件。

步驟二：修改/etc/named.conf

從 Master DNS copy 其/etc/named.conf 至 Slave DNS，然後進行修改。將 type 由 master 改為 slave，並指定 masters 為 61.219.23.88。

```
[root@dns2 ~]# scp    61.219.23.88:/etc/named.conf /etc
root@61.219.23.88's password:
named.conf                                    100% 1060       1.1MB/s     00:00
```

```
[root@dns2 ~]# vi /etc/named.conf
// generated by named-bootconf.pl

options {
        directory "/var/named";
         /*
         * If there is a firewall between you and nameservers you want
         * to talk to，   you might need to uncomment the query-source
```

```
        * directive below.   Previous versions of BIND always asked
        * questions using port 53， but BIND 8.1 uses an unprivileged
        * port by default.
        */
       // query-source address * port 53;
};

//
// a caching only nameserver config
//
controls {
       inet 127.0.0.1 allow { localhost; } keys { rndckey; };
};
zone "." IN {
       type hint;
       file "named.ca";
};

zone "localhost" IN {
       type master;
       file "localhost.zone";
       allow-update { none; };
};

zone "0.0.127.in-addr.arpa" IN {
       type master;
       file "named.local";
       allow-update { none; };
};
#下面這段文字是在前一期文章演練三所加入
zone "blue-linux.com" {
       type master;
       type slave;
       masters {61.219.23.88;};
       file "db.blue-linux-slave";
};

/* 此次演練再加入下面這段文字代表負責的網段為 61.219.23.0/24，反解 Zone
```

```
File 為 db.61.219.23 */
zone "23.219.61.in-addr.arpa" {
        ~~type master;~~
        type slave;
        masters {61.219.23.88;};
        file "db.61.219.23-slave";
};
include "/etc/rndc.key";
```

步驟三：修改/var/named 的 permission

因為 Slave DNS 是以 named 的身份做 Zone Transfer，named 這個使用者對/var/named 並無 Write 的 Permission，但做 Zone Transfer 時，Slave DNS 會將 Master 的資料寫入/var/named 目錄內，所以必須開放給 named 有 Write 的 Permission。

```
[root@dns2 ~]# ls -dl /var/named
drwxr-x---    2 root     named        4096 11 月  23 06:21 /var/named
[root@dns2 ~]# chmod g+w /var/named
[root@dns2 ~]# ls -dl /var/named
drwxrwx---    2 root     named        4096 11 月  23 06:21 /var/named
```

步驟四：啟動並檢查 Zone Transfer 是否成功

```
[root@dns2 ~]#cd /var/named
[root@dns2 /var/named]# ls
localhost.zone   named.ca   named.local
此時只有 3 個 caching-nameserver 套件所提供的 3 個檔案
[root@dns2 /var/named]# service named start
啟動 named:                                          [  確定  ]
[root@dns2 /var/named]# ls
db.61.219.23.88-slave   db.blue-linux-slave   localhost.zone   named.ca
named.local
```

多了兩個 Zone File，分別為向 Master DNS 做 Zone Transfer 得來的網域正解 Zone File「**db.blue-linux-slave**」及反解 Zone File「**db.61.219.23.88-slave**」。

作者簡介

林彥明 （ Alex Lin ）： RedHat 技術顧問，現任職於 IBM Taiwan 技術支援中心，負責 Linux、AIX、WebSphere 相關技術支援工作，具有 RHCX （RedHat

認證主考官)、RHCE、NCLP、LPIC、IBM AIX Expert、MQ、SCJP、SCWCD 國際認證，參予建置臺灣第一套商業用 IBM 1350 Linux 叢集系統。