TWNIC 93 年 DNS 教育訓練計畫

DNS 技術實作班

台灣網路資訊中心
技術組編撰

初版:2003.07.05
修訂:2004.04.28

第一章 BIND 的基本設定

第一節 基本設定練習

實作一:

請為 TWNIC 公司設定 DNS 伺服器，公司共有五部機器，其中前兩部為 DNS Server，主機名稱與 IP 對應如下:

ns1.twnic.com.tw (Master IP)

ns2.twnic.com.tw (Slave IP)

www.twnic.com.tw (211.72.100.10)

mail.twnic.com.tw (211.72.100.11)

ftp.twnic.com.tw (211.72.100.12)

請將老師電腦的 IP 設為可作 Zone Transfer，並與左右同學分別為 Master/Slave 關係，注意正反解皆須設定且一致(左同學正解為 Master，反解為 Slave；右同學相反)。

目標：復習第一天課程所學，並實際了解如何設定 Master/Slave 及 Zone File

說明:Master IP 請使用現在 PC 之 IP，Slave IP 則詢問左右同學。請注意 Master/Slave 設定上之語法差異。

範例：

建議不參考範例而直接設定，若對設定不了解之處可請教同學或講師，以便跟上進度

BIND 設定檔

```
#/etc/named.conf
options {
directory "/var/named";
allow-transfer  {老師的 IP;};
};
zone "." {
      type hint;
      file "named.ca";
};

zone "twnic.com.tw" {
type master;
allow-transfer {另外同學的 IP;};
file "twnic.com.tw.hosts";
};

zone "100.72.211.in-addr.arpa" {
type slave;
masters {另外同學的 IP;};
file "211.72.100.rev";
};
```

Options 請注意有加 s
Named 相關檔案置於 /var/named 目錄下
允許來自 老師 IP 的 AXFR 要求
注意: 每個描述要加 ; 號區隔

定義 root dns Server 的檔案, 可由
ftp://ftp.internic.net 取得或執行
dig @a.root-servers.net . ns > /var/named/named.ca

網域 twnic.com.tw
為主控網域(Master)
正解檔(FQDN->IP)為 twnic.com.tw.hosts

IP 211.72.100.X 之反解, IP 位址需要反寫
為主控網域
反解檔(IP->FQDN)為 211.72.100.rev

zone “0.0.127.in-addr.arpa” { //…localhost 設定略};

正解檔

```
# /var/named/twnic.com.tw.hosts
$ttl 38400
twnic.com.tw.     IN SOA      ns1.twnic.com.tw.  abelyang.twnic.com.tw.  (
      988077514      ; searil  number
      10800      ; refresh
      3600       ;retry
      864000     ;expire
      38400 )    ;ttl
            IN  NS  ns1.twnic.com.tw.
            IN  NS  ns2.twnic.com.tw.
ns1         IN    A      Master IP
ns2         IN    A      Slave IP
mail        IN    A      211.72.100.11
www         IN    A      211.72.100.10
ftp         IN    A      211.72.100.12
```

TTL = Time To Live

Domain IN SOA NameServer Email@admin (

序號;

更新；

重試；

過期；

有效期)

反解檔

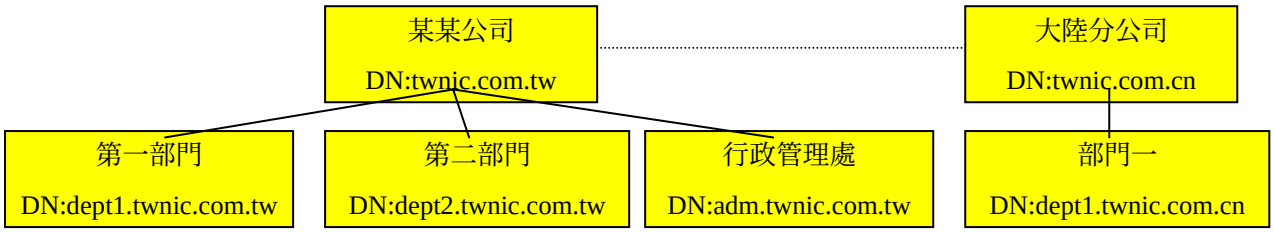
```
#/var/named/211.72.100.rev
$ttl 38400
@           IN    SOA   ns1.twnic.com.tw.  abelyang.twnic.com.tw.  (
                  988077623
                  10800
                  3600
                  864000
                  38400 )
            IN    NS    ns1.twnic.com.tw.
            IN    NS    ns2.twnic.com.tw.
10          IN    PTR   www.twnic.com.tw.
11          IN    PTR   mail.twnic.com.tw.
12          IN    PTR   ftp.twnic.com.tw.
```

以上都做好準備後，先進行檢測，可使用 BIND tool，像 nslookup 或 dig 來查詢。

第二章 DNS 功能深入介紹

第一節 正反解子網域的分割與授權

公司或學校可能分成數個部問或系所以及分公司或分部等等，此時的 DN 可能會出現依部門或系統管理的狀況:



一般狀況，若公司規模較小，上線機器較少可能較無此需求，但以規模較大公司而言，公司可能會依部份再切出子網域（Sub-Domain），而分公司可能會有自己的域名・・・ 等諸多狀況，甚至有些公司為了產品可能單獨申請一個域名皆有可能，固在 Sub-Domain 多個域名的管理工作上顯得重要許多

第二節 不切 Sub-Domain 的狀況

```
zone “twnic.com.tw” {
    type master;
    file “twnic.com.tw.hosts”;
    allow-transfer { Slave_IP;};
};
```

Zone File 的部份內容

```
$ORIGIN twnic.com.tw.
mail        IN  A   211.72.100.11
www         IN  A   211.72.100.10

$ORIGIN detp1.twnic.com.tw.
www         IN  A   140.112.55.65

$ORIGIN detp2.twnic.com.tw.
www         IN  A   140.112.55.75
```

如此，DNS 以集中的方式管理網域名稱下的所有資料，但當單位的主機有所變動時，需將需求反應給 DNS 的管理人員，請其調整 DNS 的資料。如果需求變化較多時，相對的造成 DNS Restart 的狀況及管理人員的負擔也會較多，故衡量狀況，有時我們會依其部門在 twnic.com.tw. 的網域名稱下再建立 Sub-Domain，這些 Sub-Domain 由部門自行管控

第三節 切 Sub-Domain 的狀況

```
$ORIGIN twnic.com.tw.
mail        IN  A    211.72.100.11
www         IN  A    211.72.100.10

dept1       IN  NS   ns1.dept1
dept1       IN  NS   ns2.dept1
ns1.dept1   IN  A    140.112.55.65
ns2.dept1   IN  A    140.112.55.66

$ORIGIN detp2.twnic.com.tw.
            IN  NS   ns1
            IN  NS   ns2
ns1         IN  A    140.112.55.75
ns2         IN  A    140.112.55.76
```

在部門一的 DNS 伺服器上的 Zone 即可設為“dept1.twnic.com.tw”(不可設為 twnic.com.tw.)並在正解檔中設定部門內的主機, 日後當部門一的主機有更動時, 直接修正此正解檔即可

第四節 分公司的設定方式

分公司之域名與總公司是原全不同的 ccTLD，如果分公司之域名由分公司自行管理，只要自行設定即可。但可能的狀況是分公司將域名的指定設到總公司，如此，總公司這一台 DNS 伺服器可能就會管理多個域名：

```
zone  “twnic.com.tw”   {…};
zone  “twnic.com.cn”   {…};
zone …..
```

第五節 多個 .com.tw 域名的設定方式

多個 .com.tw. 的域名，這是很平常的狀況，基本上這些域名您皆要分別設一個 Zone，不能因為偷懶而設成 .com.tw. 的 Zone ，如此會造成 Bad Referral 的狀況，因為您並不負責 .com.tw. 之域名，並無其實際指向(Delagation)，反而會造成錯誤。這個問題也常非生在反解上，如您有多個連續的 Class C (學校)，您並不能將其設成一個 Class B 的反解，道理同上。

練習：

實作二: 請左右同學以 twnic.com.tw 之域名再向下授權，並以 nslookup 檢驗結果。

如：

左同學 twnic.com.tw. ，將 dept1.twnic.com.tw. 授權給右同學管理，並檢驗雙方 NS 記錄是否一致，可否查到其他記錄(ex:www.dept1.twnic.com.tw)。

目標：了解如何實作 NS 的授權及建立 Sub-Domain。

第六節 DNS 與郵件主機

基本上 Mail Server 或 DNS 的服務相關性相當大，許多人郵件不同的狀況即是 DNS 工作不正常所致，但卻又常在 Mail Server 上找原因，究其問題大概可分成幾類：

1. 在 TWNIC 的 DNS 指定上設了郵件主機的名稱，但沒架 DNS Server
   - 從外面寄信件給您時部份會通，部分的信件會不通，出現找不到對方主機的訊息,原因在於若對方使用 BIND9 版本即會寄不到,因為沒法解析.
2. 在 DNS 指定上設了郵件主機的名稱，也架了 DNS Server ，但是 NS RR 沒設好，造成 Lame Server 狀況
   - 從外面寄信件給您時部份會通,部分的信件會不通,出現找不到對方主機的訊息,同理,主要還是因為 BIND 版本差異上的運作問題
3. 架了 DNS Server 但是沒有 MX RR
   - 這沒有太大影響,因為絕大多數的 Mail Server 並不會檢查一定要 MX, 使用 A 亦可,例如 twnic.net.tw 即是.
4. 架了 DNS Server 也有 MX RR，但是 MX RR 的 RDATA 指向一 CNAME RR
   - 退信訊息會有 loopback to me 或類似的訊息，
5. 主機沒設反解（這類問題較不多）
   - 有部分的 Mail Server 會檢查正反解或其一致性之問題
6. Mail/DNS Server 的網路連線品質不好
   - 退信會顯示 Timeout 的狀況，並說明是 Name Server 或 Mail Server timeout

第三章 BIND 細項功能設定

第一節 Options 全域項目設定

options {	options {
[directory path_name;]	zone file 的預設存放位置(def=/etc)
[named-xfer path_name;]	slave AXFR 存放的位置, (def=同上)
[dump-file path_name;]	core dump 預設位置(def=同上)
[pid-file path_name;]	named 的 PID 存放位置(def=directory 指定)
[auth-nxdomain yes_or_no;]	是否保留負面資料(def=no), 即不正確資訊的狀況是否做 Cache
[fake-iquery yes_or_no;]	作假 DNS server 的反解(def=no)
[fetch-glue yes_or_no;]	不做任何的 cache(no) , (def=yes)
[multiple-cnames yes_or_no;]	一個 FQDN 可否做 IN CNAME 多次(Round Robin 的一種)
[notify yes_or_no;]	Zone 變更通知(def=yes)
[recursion yes_or_no;]	遞迴查詢,回應問的人去哪裏查(no)(def=yes)
[forward (only | first);]	Only 只使用代詢,first 則先使用代詢
[forwarders { [in_addr ; [in_addr	代詢伺服器, 找不到使資料都往該 IP 送(若此項有值則上一個項目預設為 first)
[check-names (master | slave | res	檢查 FQDN 名稱不合法性於 type 為 (master | slave | 查詢要求)
(warn | fail | ignore);]	就 (警告|失敗|忽略)
[allow-query { address_match_list }	允許從 IP 查詢,可使用 acl_name
[allow-transfer { address_match_lis	允許從 IP AXFR ,可使用 acl_name
[listen-on [port ip_port] {	DNS 傾聽 port 為 ? IP 為 ? , 不建議更改
address_match_list };]	
[query-source [address(ip_addr|*)]	查詢外部的 DNS(IP|*)時使用 ip_Port, 不建議更改
[port(ip_port|*)];]	
[max-transfer-time-in number;]	*AXFR 的最大分鐘(def=120m)
[transfer-format(one-answer|many-an	*AXFR 時一次幾筆 RR (def=one-answer)
[transfers-in number;]	同時間最大的 AXFR(in) 數目(def=10)
[transfers-out number;]	同時間最大的 AXFR(out) 數目(def=10)
[transfers-per-ns number;]	每部 NS 同時間 AXFR 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(def=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(0~1800,0 不留,def=600)
[max-ncache-ttl number;]	負面資料的快取秒數(def=10800(3H) , N<7D)

第二節 Zone 轄區設定

Master Zone

```
zone “domain_name” {
  type master;
  file path_name;
  [ 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; ... ] };
};
```

Zone “網域名稱” {
　　類別為　　主要;//表示權威主機
　　儲存在　什麼檔案中;// 即 zone file
　　允許來自 IP 的動態動新(使用 nsupdate 指令)
　　轄區資料變更是否同通知 Slave 主機(NS RR)
　　轄區資料變更時通知那些 DNS 主機(IP)
註: zone 之內容略有刪減, 去除的為不重要或甚少使用的項目

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; ... ] };
  };
```

Zone “網域名稱” {
　　類別為 次要;//表示備份主機
　　儲存在 什麼檔案中;// 即 zone file
　　主要的 zone 在 port 多少 { IP ;};

Forward Zone

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

第三節 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;
    DN;
    path_name;
    CIDR;
    None;
    Any;
    Localhost;
    Localnets;
```

IP 位址 IP/[netmask]
網域名稱 *.twnic.net.tw
檔案名稱, 內存 ACL
IP 段211.72.210/23
沒有任何 IP
任何 IP
localhost (127.0.0.1)
網卡的 IP/Netmask (即相連的網路)

```
//CIDR 簡介可參考 http://www.microsoft.com/taiwan/TechNet/top10/top04_11.htm
//其他較不重要項目可查 man named.conf
    };
```

例如:

```
    acl TWNIC {211.72.210.0/23;210.17.9/24;}
```

在別的“參數”再定義其行為

```
    allow-transfer { TWNIC;};
    allow-query { TWNIC;};
    …
```

第四節 logging 系統訊息的設定

若您不定義 logging 並不會影響您系統的運作，而在不定義的狀況下，所有的系統記錄皆會送給到 syslog 中的 daemon.*，通常這個檔會是 /var/log/messages，裏面存放著所有 daemon 的系統記錄，會顯得過於雜亂，故透過 logging 除了幫我們做分類外，尚可將一些重要的訊息 log 下來，以便更進一步分析

```
logging {
  [ channel channel_name {
    ( file path_name
[ versions ( number | unlimited ) ]
[ size size_spec ]
      |syslog(kern|user |mail |daemon |auth |syslo
  news | uucp | cron | authpriv | ftp |
  local0 | local1 | local2 | local3 |
  local4 | local5 | local6 | local7 )
     | null );
    [ severity ( critical | error | warning | not
   info  | debug [ level ] | dynamic ); ]
    [ print-category yes_or_no; ]
    [ print-severity yes_or_no; ]
    [ print-time yes_or_no; ]
  }; ]
  [ category category_name {
    channel_name; [ channel_name; ... ]
  }; ]
  ...
};
```

開始 logging 節段

設定 通道 通道名稱 可

要記錄的檔案名稱

要檔案備份數目

每個備份大小 或

轉成 Syslog 的格式(可參考 man syslog)

或

沒有

記錄的等級 (debug 為除錯等級,dynamic 則是視 USR 訊號而定)

記錄檔中要輸出 category 名稱

severity 名稱

時間

類別 類別名稱 { 通道名稱 };每個 BIND 版本稍有不同

default	指定以外的所有訊息
cname	CNAME 的錯誤
lame-servers	不良的委任設定
load	載入 zone file 時的訊息
notify變更通知	
os	作業系統相關問題
packet	收送之封包,需指向檔案
panic 引起	dns 關閉的原因
parser	檢查組態檔的語法
statistics	DNS 活動的定時報告
security	被認可或不被認可的請求
update	動態更新事件
xfer-in	zone-transfer in
xfer-out	zone-transfer out

範例

```
logging {
  channel default_log {
    file "/var/log/named/dns-default.log" versions 10 size 1m;
    severity info;
  };
  channel lamer_log {
    file "/var/log/named/dns-lamer.log" versions 3 size 1m;
    severity info;
    print-severity yes;
    print-time     yes;
    print-category yes;
  };
  channel query_log {
    file "/var/log/named/dns-query.log" versions 10 size 10m;
    severity info;
  };
  channel security_log {
    file "/var/log/named/dns-security.log" versions 3 size 1m;
    severity info;
    print-severity yes;
    print-time     yes;
    print-category yes;
  };
category lame-servers   { lamer_log; };
category security{ security_log;};
category queries { query_log;};
category default {default_log;};
};
```

練習：

實作三：請試做一 named.conf，從最能增加安全的角度著眼

目標: DNS 的安全日益重要，此處的練習是讓同學了解如何設定一個正確且安全的 DNS 系統。

說明： 本題並無特定答案，詳細狀況可參見附錄一及附錄二，並建議您可多了解 DNS 安全之章節說明。

第四章 BIND9 的新功能介紹

第一節 BIND9 與 BIND8 之差異

	Windows	BIND
操作	❑GUI的設定方式入門容易 ❑可用 Windows 其他服務結合 (WINS/AD)	❑設定以文字編輯進行 ❑較易出錯 ❑Unix 環境為一般人所不熟悉
效率	❑查詢數字無統計資料	❑每秒可處理上萬次查詢 ❑Multi-thread
穩定性	❑視 OS 表現 ❑基本上可符合一般企業需求	❑穩定性佳 ❑版本更新速度較快
安全性	❑隨系統版本更新而更新版本	❑可從設定面加強安全性 ❑較能預防DNS Spoofing
佔有率	❑在台灣兩者相當	❑在全世界佔大宗
其他		❑Root Server 皆以 BIND 為主

第二節 對 IPv6 的支援

IPv6 位址格式的說明

IPv6 由 16 個 bytes 所組成，固定位址表示可達 2^128 次方，較原來 IPv4 4 bytes 2^32 次方的長度多出無數倍，但也因為長度變長了，其位址表示也異於 IPv4

IPv6 表示法 2001:238:882:0:0:8bff:fedd:df6/48

- 以二個 bytes 為一組，並以 HEX 方式表示 如 2001 為 16 進位值
- 每組間以 : 區隔，故共會有八組，合計為 16 bytes
- 若組內有為 0 之資料可以 :: ，如上例可成為 2001:238:882:::8bff:fedd:df6/48
- 每個 IPv6 位址只能用一次 :: 省略
- /48 為 prefix ，即網路遮罩，IPv6 皆以 CIDR 方式表示為主
- 如此長的位址表示，顯示 DNS 的功能更形重要

IPv6 的 DNS 設定

正解 A/AAAA RR

原來 DNS 設定 Address 記錄時以 A RR 來表示:

www.twnic.net.tw. IN A 210.17.9.228

而 IPv6 就長度而言較 IPv4 多四倍，故其在 DNS 中的 A RR 可以 AAAA RR 來表示

www.twnic.net.tw. IN AAAA 2001:238:882::8bff:fedd:df6

如果這兩筆記錄皆存在我們的 DNS 中，其取用的條件仍是依照原來 DNS 查詢之原則，如查詢 A RR 或 AAAA RR 而定，如果您具有 IPv6 的環境，原則上根據 OS 定義，其會先查詢 AAAA RR 取用，若無則取 A RR。

正解 A6 RR

IPv6 具有 Renumbering 之功能，即是當您動態配置 IP 或更換 ISP 時，其所更改之 Address 為您的 NetID，而 HostID 並不會改變，如您的 IPv6 位址:

2001:238:882::8bff:fedd:df6/48

當您更換 ISP 時，會改變的是 *2001:238:882/48* 這一段，而後半段並不需要改變，因應這個特性，DNS 亦定義 RR 為 A6 之項目:

$ORIGIN twnic.net.tw.
www 3600 IN A6 48 ::8bff:fedd:df6 host1.ISP1.net.tw.
$ORIGIN ISP1.net.tw.
host1 3600 IN A6 0 2001:238:882::

::8bff:fedd:df6，而前 64 bits 則要問 host1.ISP1.net.tw. 之答案，而得到結果，而組成其原來之 IPv6 之 128 bits。

以一般 IPv4 的狀況而言，更換 ISP ，IP address 亦要根著改變，而 DNS 的指定或設定內容亦要根著改變，這樣的工作是相當累人的，因為網路狀況的變動對公司的運作有莫大的影響。IPv6 + DNS A6 RR 則可大量的避免此類的狀況，因為您僅需要調動 host1.ISP1.net.tw. 之值到 ISP2 即可完成整個設定

$ORIGIN ISP2.net.tw.
host2 3600 IN A6 0 1234:5678:90ab::

反解 Nibble Format

即是一般我們習慣的反向寫法:

IPv4: 210.17.9.228 => 228.9.17.210.in-addr.arpa.

IPv6:

2001:238:882::8bff:fedd:df6=>6.f.d.0.d.d.e.f.f.f.b.8.0.0.0.0.0.0.0.0.2.8.8.0.8.3.2.0.1.0.0.2.ip6.arpa.

如此我們可以發現,IPv6 的反解真是又臭又長,且因為反過來寫而容易筆誤,故 BIND 9 衍生出一種名為 Binary Format 之格,即可照原來排列方式。

反解 Binary Format

IPv6:

2001:238:882::8bff:fedd:df6=>[\0x2001023808820000000008bfffedd0df6].ip6.arpa.

請注意補 0 及 :: 間省略之問題。上面的表示法較 Bibble Format 基本上來的簡單且好用許多。

第三節 新增功能

變數 $GENERATE

$GENERATE 用於 Zone File 中,如同我們使用 $ORIGIN、$TTL 等,但 $GENERATE 基本上是產一如同一迴圈之作用,讓我們的 Zone File 可以更簡捷些,原來我們設定一 210.17.9.x Class C 之反解,可能在 Zone File 內會如此寫:

```
$ORIGIN 9.17.210.in-addr.arpa.
1    IN  PTR     pc1.twnic.net.tw.
2    IN  PTR     pc2.twnic.net.tw.
3    IN  PTR     pc3.twnic.net.tw.
...
255  IN  PTR     pc255.twnic.net.tw.
```

但使用 $GENERATE 則可簡略成一行

```
$ORIGIN 9.17.210.in-addr.arpa.
$GENERATE 1-255 $   PTR     pc$.twnic.net.tw.
```

$GENERATE 可以使用於正解或反解,並且 1-255 可指定任意合理範圍,是一相當實用之功能變數。在 BIND 9 中,class IN 是可寫可不寫的,但在使用 $GENERATE 時,不能使用 IN 在字串中,這是其特性。

$GENERATE 不能做運作處理,且亦不能用巢狀迴圈表示,最多僅能用 1-255/10 表示 1、11、21…等間隔 10 之特性。

新增 view 的功能

```
view "view_name" {
        match-clients { address_match_list };
        [ view_option; ...]
        [ zone_statement; ...]
};
view "view_name1" {
  match-clients { 211.72.210/24; };
  recursion yes;
  zone "twnic.net" {
        type master;
      file "twnic.net.hosts";};
};
view "view_name2" {
  match-clients { 211.72.211/24; };
  recursion no;
  zone "nic.net.tw" {
  type master;
  file "nic.net.tw.hosts";};
 };
```

View 的功能讓您設定不同的 IP(match-clients) 有不同的功能，如範例 view_name1 所定義的 match-clients 才能查得到 twnic.net 網域名稱之資訊，且其查詢為 recursion。最常使用之處為當我們使用 NAT 環境時,內部查詢某些 FQDN 時,我們可能希望結果為 Private IP, 而不是外面人來查的結果,在 BIND 8.X 版本時,會建議要建兩個 DNS Server , 一個服務外部的查詢,一個則專門對內,但如此無形中即增加的成本及維護上的問題,故,我們可以在這個時候使用 view 的功能,只要能夠區分來源 IP , 即可控制其取得不同的解析結果

新增 rrset-order 功能

```
rrset-order {   [ class class_name ][ type type_name ][ name "domain_name"]
order ordering };  //ordering 可為 fixed/random/cyclic
```

```
rrset-order {class  IN  type    A   name "twnic.net.tw";    order random;};
rrset-order { class ANY type    ANY name "*";               order cyclic ;};
```

rrset-order 多筆 FQDN 時回應的方法，也就是當您有 Round Robin 的資料(一個名稱對應到多個 RDATA)時，其回應的方式:

random: 隨機選取，原來之系統預設值

cyclic: 循環式回答，即有三筆資料的話，即依 1->2->3->1->2…回答

練習：

實作四：如何以一部 DNS 伺服器實現不同 IP 來源(例如：內部網路/外部網路)時有不同的查詢結果

目的:　實際練習 View 之應用

說明：　請實作 view 之功能，讓另外一位同學來查 www.twnic.com.tw 時，給予其答案為 192.168.0.100，由其他位置過來查則得到原始 IP(211.72.100.10)，並以 dig 來檢測。

第四章 DNS 維護與管理

第一節 系統記錄分析

DNS 的系統記錄相同複雜,一般而言建議您使用 logging 功能將系統記錄做一分類以方便您查閱

以下茲對常見的訊息做一些介紹，詳細說明可見於下頁:

Jun 11 11:20:48 twnic.net.tw named[6103]: ns_forw: query (mail.vinwell.com.tw.vinwell.com.tw) All possible A RR's lame

- **說明這個網域名稱可能有 Lame Server 狀況，也有可能是查詢人之 Resolver 之 default-domain 之功能所引起的**

Jun 11 14:03:49 twnic.net.tw named[19709]: /etc/named.conf:53: syntax error near '}'

- **語法錯誤，通常是少了 ; 號或忘了 {} 號，可使用 named-checkconf 檢查看看**

Jun 11 11:20:02 twnic.net.tw named[6103]: unrelated additional info 'Manager.aluba.com.tw' type A from [203.149.224.202].53

- **說明從 203.149.224.202 送來了和他無關的資料，通常發生的原因可能是該部 DNS 的一些域名設定有誤，但也有可能是 DNS 欺騙的一種狀況**

4. Jun 11 11:26:32 twnic.net.tw named[6103]: bad referral (in-addr.arpa !< 88.216.1.in-addr.arpa) from [168.95.192.14].53

- **Bad Referral 狀況，可以從訊息中看到 168.95.192.14 (HINET)的反解設為 "in-addr.arpa." ，可能原因是其 IP 太多，故如此偷懶，這樣的設定會造成此人往後的一段時間內($TTL) 查詢任何反解資料時都會到此 IP 查**

Jun 11 11:28:59 f.dns.tw named[10796]: ns_forw: query(BBS\.NNUT\.EDU\.TW\000.nhu.edu.tw) forwarding loop (dns.nhu.edu.tw:203.72.0.3) learnt (A=140.111.1.2:NS=140.111.1.2)

- 查詢 bbs.nnut.edu.tw. 時發現有迴圈狀況，通常是 CNAME 所造成的，同時並記住管理 nhu.edu.tw. 的上層主機為 140.111.1.2（也就是這一次是使用這一部）

Jun 11 11:36:07 twnic.net.tw named[6103]: Response from unexpected source ([168.95.1.24].53) for query "_ldap._tcp.tmt.gtmt.com.tw IN SOA"

- 查詢 _ldap…主機時,意外的從 168…的主機回應答案,其主要原因是 HINET 使用 Layer4 Switch，故回應的 IP 可能不固定所致，但亦有可能是 DNS Spoofing

Jun 11 11:55:06 twnic.net.tw named[6103]: Cleaned cache of 1083 Rrsets

- 這是 named 清楚 Cache 的 log，表示有 1083 筆 TTL 時間到了

Jun 11 11:58:18 f.dns.tw named[10796]: rcvd NOTIFY(tw, IN, SOA) from [140.111.1.2].45599

- 從 140.111.1.2 將到一個轄區變更的通知 (notify)

Jun 11 13:09:11 b.dns.tw named[23410]: Sent NOTIFY for "tw IN SOA 2001021959" (tw); 4 NS, 4 A

- 同上，但是送出，可見送出時所帶參數，2001… 為序號

Jun 11 15:32:23 pc071 named[480]: sysquery: findns error (SERVFAIL) on pc071.twnic.net.tw?

- 這一部 DNS 發生錯誤造成 伺服器失敗

Jun 11 12:39:14 twnic.net.tw named[6103]: sysquery: query(mail.digi-age.com.tw)NS points to CNAME (lnx5.net-chinese.com.tw:) learnt (CNAME=168.95.192.3:NS=140.111.1.2)

- 查詢 mail.digi…時，發現它的 NS RR 接的是一個 CNAME RR，這樣的 NS 是不被承認的

Jun 11 13:56:54 twnic.net.tw named[19251]: db_load could not open: /var/named/hosts: No such file or directory

- zone 的設定中指定的 file 位置找不到檔案

Jun 11 13:08:51 b.dns.tw named-xfer[30088]: send AXFR query 0 to 168.95.192.9

- 這部機器向 168.95.192.9 作一 AXFR 請求，這個請求送的序號為 0 (因為這是使用 named-xfer)

Jun 11 13:08:52 f.dns.tw named[10796]: slave zone "tw" (IN) loaded (serial 2001021959)

- 這部機器完成了”tw” 的 AXFR 請求，並將其載入

Jun 12 01:54:15 www named[11676]: check_hints: no A records for E.ROOT-SERVERS.TW class 1 in hints

- 找不到 Root Server 中 E.ROOT…之 Class，也就是 named.ca 中沒有設定 IN

Jun 12 04:07:54 a.dns.tw named[19468]: stream_getlen([140.126.102.30].42873): Connection timed out

- 和 140.126.102.30 的連線逾時

Jun 12 09:29:19 u10 named[25400]: denied update from [140.112.15.188].3633 for "."

- 拒絕來自 140.112.15.188 要求動態更新 . 的位置

Jun 12 09:42:58 registry named[255]: check_hints: root NS list in hints for class 1 does not match root NS list

- named.ca 中沒有 NS RR

Jun 12 15:44:51 pc071 named[17429]: ?萎辣.蝳~N 鋓 Q~F.tw has multiple CNAMES
- 這個名稱 CNAME 多次到不同的名稱(系統預設是 no)

Jun 12 18:00:08 u10 named[25400]: Zone "" (file named.ca): No default TTL ($TTL<value>) set, using SOA minimum instead
- named.ca 沒有定 TTL 值，使用 SOA 中的 TTL 代替

Jun 12 18:00:08 u10 named[25400]: a.dns.tw IN A differing ttls: correct
- a.dns.tw 的 TTL 值與上層設定不同，調整之

Jun 12 18:58:29 pc071 named[24909]: /var/named/abel.hosts.utf8: WARNING SOA refresh value is less than 2 * retry (2 < 1209600 * 2)
- SOA 中的 refresh 的值小於 retry 值的二倍，建議您可達十倍左右

Jun 16 08:12:50 f.dns.tw named[10796]: sysquery: findns error (NXDOMAIN) on ipopc-16.ipoware.ocm.tw?
- 找不到 這個網域名稱 (NXDOMAIN)

Jun 19 12:01:06 twnic.net.tw named[20004]: Malformed response from [203.70.159.194].53 (out of data in final pass)
- DNS 回應的封包格式不對

Jun 8:58:01 pc071 named[24901]: deleting interface [211.72.211.71].53
- 這個網路介面 shutdown 了

Jun 12 18:58:04 pc071 named[24901]: There may be a name server already running on [211.72.211.1].53
- named 已經在執行中了

Jun 26 06:11:59 twnic.net.tw named[20004]: invalid RR type 'NS' in additional section (name = 'auth.com') from [194.74.63.90].53
- 194…的回答中，additional section 資訊中的 NS 資料不對，可能有設錯的狀況或 Spoofing

Jun 21 17:14:44 pc071 named[6884]: log_open_stream: open(/var/log/named/dns-statistics.log) failed: Permission denied
- 開啟系統記錄檔時，權限不足

Jun 21 17:14:59 pc071 named[6884]: cannot set resource limit on this system
- 無法設定這台機器的系統設定，如最大檔案開啟數，最大行程數・・・

Jan 7 13:58:01 pc071 named[231]: db.movie:16: data "hp.com" outside zone "com.tw" (ignored)

- hp.com 這個名稱不屬於 db.movie 這個檔案的轄區

Jun 11 12:00:00 pc071 named[17898]: zone "tw" (class 1) SOA serial# (1) rcvd from [211.72.210.250] is < ours (2001062901)

- pc071 轄區傳送 "tw" 時發現 211.72.210.250 的序號小於現在的序號, 這會造成不傳的狀況. 每次更改 zone file 時要記得同時至少為 serial 加上 1, 這是一個基本的原則

Jan 6 11:55:25 pc071 named[544]: Err/TO getting serial# for "edu.tw"
Jan 8 17:12:43 pc071 named[22261]: secondary zone "edu.tw" expired

- slave 無法取得轄區資料,當到達了 SOA 的 expire 值時則會產生第二行的訊息內容, 表示 edu.tw 的轄區資料已經過期了, 當再遇到有人來查 twnic.edu.tw 時則會產生 SERVFAIL 的系統訊息.會造成這樣的訊息可能的原因可能為: master 與 slave 間的網路連線有問題, slave 主機所指定的 master ip 有誤, master 沒有開 allow-transfer 或其 zone file 中的語法有誤

Jan 6 11:59:29 pc071 named[544]: can't change directory to /var/named: No such file or directory

- 找不到 /var/named 這個目錄，這通常是有 directory 所指定的

Jan 6 15:07:46 pc071 named[693]: master zone "movie.edu" (IN) rejected due to errors (serial 1997010600)

- Zone file 中語法或格式錯誤造成整個 Zone 不用

Dec 12 11:52:11 pc071 named[7770]: socket(SOCK_RAW): Too many open files

- 系統開啟的檔案數過多，造成 named 無法再開啟檔案或 socket

Sep 24 10:40:11 pc071 syslog: gethostby*.getanswer: asked for "37.103.74.204.in-addr.arpa IN PTR" , got type "CNAME"
Sep 24 10:40:11 pc071 syslog: gethostby*.getanswer: asked for "37.103.74.204.in-addr.arpa", got "37.32/27.103.74.204.in-addr.arpa"

- 找 204.74.103.37 的反解時，發現其指到一 CNAME，通常這種狀況較少見，是小於一個 Class C 中常用反解的指定方法，所以其要再問 37.32/27.103.74.204.in-addr.arpa 的結果

Sep 24 10:40:11 pc071 named[7770]: ns_udp checksums NOT turned on: exiting

- OS 中的 udp checksum 功能未打開，DNS 無法運作

BIND 的系統記錄非常多，並不同的版本可能會稍有不同，由於資料的搜集不易且敝中心多使用 BIND 9.X 的環境，故無法一一詳列，若您有這方面的疑問可寫信詢問我們 (service@twnic.net.tw)。 或至 http://www.menandmice.com/docs/named_messages.htm 可查到所有的訊息及解釋。

第二節 DNS 流量分析

使用 BIND 自身之 logging 分析(僅適用 8.X 版本)

logging 中有一 category statistics 之設定有助於流量之分析

DNS 的統計資料

DNS 統計資料(系統預設 60 分鐘產生一次,可由 options 中去改變)

1. Jun 11 11:55:06 twnic.net.tw named[6103]: USAGE 992231718 991814118 CPU=57.23u/22.29s CHILDCPU=0u/0s
2. Jun 11 11:55:06 twnic.net.tw named[6103]: NSTATS 992231718 991814118 0=283 A=148995 NS=705 CNAME=6 SOA=56120 PTR=35568 MX=32875 TXT=3 AAAA=37 SRV=7456 AXFR=1 ANY=53843
3. Jun 11 11:55:06 twnic.net.tw named[6103]: XSTATS 992231718 991814118 RR=136227 RNXD=50282 RFwdR=97219 RDupR=451 RFail=3940 RFErr=0 RErr=344 RAXFR=1 RLame=7479 ROpts=0 SSysQ=34959 **SAns=261699** SFwdQ=147011 SDupQ=221625 SErr=0 **RQ=335892** RIQ=0RFwdQ=147011 RDupQ=11948 RTCP=2352 SFwdR=97219 SFail=12 SFErr=0 SNaAns=183725 SNXD=69778 RUQ=0 RURQ=0 RUXFR=1 RUUpd=0

USAGE: 目前系統時間(以秒計) named 啟動的時間 CPU 的 user mode/ system mode 的時間, 及子行程相同的資料(992231718 - 991814118 = 417600 秒(named 執行的時間))

NSTATS: 每個資源記錄的查詢次數(A,NS,CNAME,SOA,PTR,MX,TXT,AAAA,SRV….)

XSTATS: 詳見下一節的欄位說明

平均查詢次數:

RQ/417600=0.8 (從 named 啟動以來每秒收到約 0.8 個查詢)

小時查詢次數:

因其為累計值，故欲計算小時的查詢，需有兩個小時的 log

Jun 11 12:55:05 twnic.net.tw named[6103]: XSTATS 992235318 991814118 RR=138360 RNXD=50842 RFwdR=98669 RDupR=466 RFail=4001 RFErr=0 RErr=344 RAXFR=1 RLame=7546 ROpts=0 SSysQ=35545 **SAns=266536** SFwdQ=148831 SDupQ=223465 SErr=0 **RQ=341387** RIQ=0RFwdQ=148831 RDupQ=12099 RTCP=2372 SFwdR=98669 SFail=12 SFErr=0 SNaAns=187447 SNXD=70533 RUQ=0 RURQ=0 RUXFR=1 RUUpd=0

(341387-338592) / (3600 秒)= 0.8 (即表示 twnic.net.tw 在一小時內收到了 2975 個查詢, 約等於每秒 0.8 次)

流量(bps):

查詢及回應各一個封包而每個封包平均大小為 800 bits

((RQ1-RQ2) +(SAns1-SAns2)) * 800 / 3600

查詢次數(11 點值-12 點值) 回應次數 平均每個 DNS 封包約 800 bits 每秒

((341387-338592) +(266536-261699)) * 800 /3600

=(2975 + 4837) * 800 /3600

=1736 bps

(可至 http://www.dns.net/dnsrd/tools.html 上有許多 bind 的分析工具可以幫您解決上述眾多欄位的問題)

named 統計資料中欄位的意義

使 named 程式馬上產生 使用狀況的列表

kill -ILL named_PID // 將產生如下列表(named.stats 檔案)

+++ Statistics Dump +++ (993612238) Wed Jun 27 11:23:58 2001

1375116	time since boot (	Named 的服務執行了多少的時間
1375116	time since reset	自從收到 HUP 訊號後執行的時間
262	Unknown query typ	不知道型態的查詢
517137	A queries	A 記錄的查詢
1617	NS queries	NS
0	MD queries	MD 及 MF 皆有 MAIL 用(類似 MX 功能), 以下未介紹的訊息表示其不甚重要或很少使用, 若您想知道其意義可參考 /usr/include/arpa/nameser.h 檔
0	MF queries	
0	CNAME queries	CNAME 查詢
157627	SOA queries	SOA 授權資料的查詢
2	MB queries	
0	MG queries	
0	MR queries	
0	NULL queries	
0	WKS queries	WKS 為存放一個字串, 為表示該 zone 中的 Well Known Service
141292	PTR queries	PTR 為反解的查詢
0	HINFO queries	HINFO 為主機資料, Host INFOrmation
0	MINFO queries	
114502	MX queries	MX 郵件路由的資訊 (Mail eXchange)
22	TXT queries	TXT 該 zone 中的一些說明資訊
0	RP queries	RP 該 zone 的負責人資訊 (Responsible Person)
0	AFSDB queries	
0	X25 queries	
0	ISDN queries	
0	RT queries	
0	NSAP queries	
0	NSAP_PTR queries	SIG/KEY 為 DNS 的安全設定, 但很少人使用
0	SIG queries	
0	KEY queries	
0	PX queries	
0	GPOS queries	
718	AAAA queries	AAAA 定義 IPv6 的 A 記錄
9	LOC queries	
0	NXT queries	
0	EID queries	
0	NIMLOC queries	
20373	SRV queries	SRV DNS 伺服器的查詢
0	ATMA queries	
0	NAPTR queries	NAPTR e.164 number 使用, 電信業必需了解(RFC 2915,2916,3026)
0	KX queries	
0	CERT queries	

```
0       A6 queries
0       DNAME queries
0       OPT queries

++ Name Server Statistics ++
(Legend)
        RR      RNXD    RFwdR   RDupR   RFail
        RFErr   RErr    RAXFR   RLame   ROpts
        SSysQ   SAns    SFwdQ   SDupQ   SErr
        RQ      RIQ     RFwdQ   RDupQ   RTCP
        SFwdR   SFail   SFErr   SNaAns  SNXD
        RUQ     RURQ    RUXFR   RUUpd
(Global)
        96 16 59 0 5  0 2 1 0 0  31 67 40 15 0  77 0 40 2 2  59 0 0 8 24  0 0 0 0
//位置     1  2  3  4 5  6 7 8 9 0  1  2  3  4  5  6  7 8 9 0  1 2 3 4  5  6 7 8
9
```

以下（Legend）之解釋

RR　　收到的回應次數，可以除以秒數可知系統查詢的頻率

RNXD　收到的 no such domain 回應次數，代表有多少次的查詢沒有相關的 DN，如果過多則需了解是否為系統的問題

RFwdR　收到回應(RR)再傳回最初查詢之處的次數

RDupR　收到的重覆查詢的次數，會多的原因為 Client 將 nameserver 指向這一台 DNS 所致

RFail　SERVFAIL 的次數，表示在詢問別的 NS 時發生了多少次的伺服器失敗的回應，應注意是那邊的系統問題(SERVFAIL 的原因多為 zone file 的 db 檔語法有錯，NS 的記憶體配置失敗,或轄區資料過期(expire)的狀況

RFErr　收的的格式錯誤(FORMERR)的訊息，在查詢時 NS 認為查詢有格式錯誤的情形發生，可使用 check-names 設定來降低此值

RErr　SERVFAIL 和 FORMERR 以外的錯誤次數，如 NXDOMAIN, REFUSED 等 (/usr/include/arpa/nameser.h 檔中有諸多定義)

RAXFR　收到 AXFR 請求的次數，實際的來源可檢看 logging 的檔案的 security 項

RLame　本機在查詢時發生不當委任的次數，實際的來源可檢看 logging 的檔案的 lame 項

ROpts　收到帶有 IP options 的封包次數，即 IP 封包中帶有 option 區段

SSysQ　系統查詢次數，由 DNS 自己發出的查詢，發生在找 ROOT DNS 及 NS 的 A 記錄

SAns　送出的回應次數,RQ=77，而回 SAns 為 67 表示有 10 次查詢石沈大海(沒有回應).

SFwdQ　送出的轉送查詢次數

SDupQ　送出的重覆查詢次數

SErr　送出查詢時發生錯誤(即 snedto() 的即統呼叫失敗)

RQ　收到的查詢次數

RIQ　收到的反解查詢次數

RFwdQ　收到的轉送查詢次數

RDupQ　收到的重覆查詢次數

RTCP　收到使用 TCP 查詢的次數(DNS 查詢一般使用 UDP)

SFwdR　所送出轉送的回應次數
SFail　送出查詢時發生 SERVFAIL 的次數
SFErr　送出查詢時發生 FORMERR 的次數
SNaAns　送出的非權威的回應次數, 67 筆的回應(SAns) 中, 有 8 (SNaAns)是快取資料
SNXD　送出的 “No Such Domain” 的訊息
RUQ　收到不認可的查詢(不在 allow-query 的範圍內)次數
RURQ　收到不認可的遞迴查詢(recursion yes)次數
RUXFR　收到不認可的 AXFR 請求(不在 allow-transfer 的範圍內)
RUUpd　收到不認可的動態更新要求(不在 allow-update 的範圍內)
解譯:

Ans	answer	回答
Dup	duplicate	重覆
Err	error	錯誤
Fwd	forward	轉送
I	iquery	反解
Lame		不良的委任(原意:跛腳)
NXD	not exist Domain	網域名稱不存在
Na	Non-authoritative	非權威
Q	query	查詢
R	receive/response	收到/回應
S	send	送出
U	unapproved	不被認可的
Upd	update	更新
F	format	格式
XFR/AXFR/IXFR		轄區傳送

基本上,上述有許多描述您並不需要,有興趣了解就看看,跳過對您亦不會有太大的影響。透過 syslog 方式來看流量可能較不具效益,故我們可以使用 signal (kill)的方式來實現

```
# DNS 流量, BIND 8.x 程式範例
#/bin/sh
killall -ILL named
cat /var/named/named.stats  | tail -3 |head -1 | awk '{print $16" "$12}'
```

這 “killall -ILL named” 我們需固定時間做,故要加入排程中(方法見下一節),而第二行則是取出 RQ/RAns (查詢/回應) 的值,是否要乘以 bits 的值,完全看您的需求,詳細做法您可參閱 mrtg 的使用方法,或下一節的部份。

使用 rndc 分析(僅適用 9.x 版)

ndc 在 8.X 及 9.X 版本皆存在，但若用於查詢流量分析的話僅有 9.X 適用。設定 ndc 但的功能是一件麻煩的事，其事前準備動作可能較多，但事後則可較輕鬆管理。

```
[root@pc071 etc]# rndc-confgen
# Start of rndc.conf 設定一 rndc.conf ，建議與 named.conf 同一目錄
key "rndc-key" {
        algorithm hmac-md5;
        secret "lnWw5u3J1r/8jaTpjfWF6w==";
};

options {
        default-key "rndc-key";
        default-server 127.0.0.1;
        default-port 953;
};
#在 named.conf 中加入如下設定
 controls {
       inet 127.0.0.1 port 953
               allow { 127.0.0.1; } keys { "rndc-key"; };
```

在啟動 DNS 後，即可使用 rndc 來檢測系統

```
[root@pc071 etc]# rndc
Usage: rndc [-c config] [-s server] [-p port]
        [-k key-file ] [-y key] [-V] command

command is one of the following:

  reload        Reload configuration file and zones.
  reload zone [class [view]]
                Reload a single zone.
  refresh zone [class [view]]
                Schedule immediate maintenance for a zone.
  reconfig      Reload configuration file and new zones only.
  stats         Write server statistics to the statistics file.
  querylog      Toggle query logging.
  dumpdb        Dump cache(s) to the dump file (named_dump.db).
  stop          Save pending updates to master files and stop the server.
  halt          Stop the server without saving pending updates.
  trace         Increment debugging level by one.
  trace level   Change the debugging level.
  notrace       Set debugging level to 0.
  flush         Flushes all of the server's caches.
  flush [view]  Flushes the server's cache for a view.
  status        Display status of the server.
  *restart      Restart the server.
```

就本章主題而言，我們要了解的是流量，故可使用下列指令來看:

```
rndc -c /etc/rndc.conf -s localhost stats
```

此時會產生三個檔案於 directory 的目錄裏，我們就其中的 named.stats 來看其內容

```
[root@pc071 named]# cat named.stats
+++ Statistics Dump +++ (1050394961)
success 4087            #查詢成功的量
referral 0              # bad referral 的量
nxrrset 0               #找不到記錄的量
nxdomain 56421          #找不到網域名稱
recursion 6765          #遞迴查詢次數
failure 6722            #失敗的量
--- Statistics Dump --- (1050394961)
+++ Statistics Dump +++ (1050395022)
success 4087
referral 0
nxrrset 0
nxdomain 56423
recursion 6765
failure 6722
```

--- Statistics Dump --- (1050395022)

上述例子中,Statistics 中間的值為每一次 stats 之狀況,而我們若欲有精確而有效的查詢與回應統計,除了時間因素要考量外,尚需注意這些值的意涵:

查詢數:全部數值總合

回應數:除 failure 外之總合

上述的格式是固定的,故我們可以準備一套方法來產生其流量資料,為了和下一節配合,我們可以產生 mrtg 格式之記錄

```
#dns-flow.sh
rndc -s localhost -c /etc/rndc.conf stats     #產生 namd.stats 供統計用
sum=0
for c in `cut -f 2 -d' ' /var/named/named.stats | tail -7 | head -6`      #取得數值部份
do
sum=`expr $sum + $c`          #加總
done
echo $sum                  #stdout 供 mrtg 使用
sum=0
for c in `cut -f 2 -d' ' /var/named/named.stats | tail -7 | head -5`
do
sum=`expr $sum + $c`
done
echo $sum
```

上述僅是一個簡短的 Shell Script,我們可以將其加到排程中,讓其固定時間可以產生結果:

```
crontab -e
```

進入編輯畫面後,填寫排程

```
# 分  時  日  月  星期  指令 ( /5 表示每5分鐘)
  */5 *   *   *   *     sh dns-flow.sh      #這個結果並未做處理,會以 mail 寄到
                                            #user 的信箱
```

使用 mrtg 產生統計圖檔

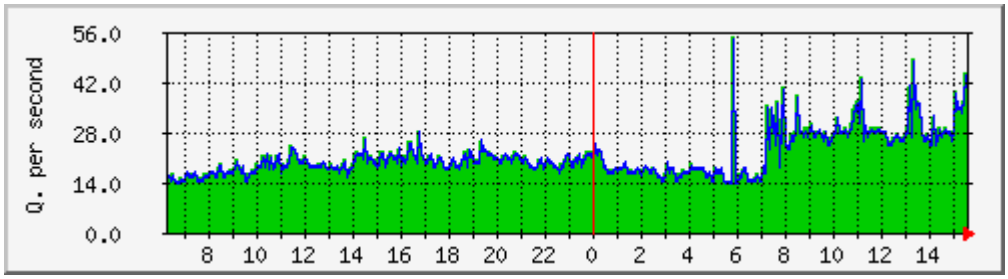
如果您上面都了解了，這一節的設定其實就會很簡單，本節並教授 mrtg 的安裝，僅列出 mrtg 的設定供您參考，詳細 mrtg 的設定您可用搜尋引擎找到許多，或直接參考官方網站 http://www.mrtg.org。

```
#/usr/local/etc/mrtg
#…前略
Target[twnic.com.tw]:‘sh dns-flow.sh‘
MaxBytes[twnic.com.tw]: 2500
Title[twnic.com.tw]: twnic.com.tw
Legend1[twnic.com.tw]: DNS 查詢(次數/秒)
Legend2[twnic.com.tw]: DNS 回應(次數/秒)
LegendI[twnic.com.tw]: DNS 查詢
Legend0[twnic.com.tw]: DNS 回應
YLegend[twnic.com.tw]: Q. per second
PageTop[twnic.com.tw]: <h1>twnic.com.tw</h1>
# ..後略
```

搭配 mrtg 及 dns-flow,我們可以定一個更完整的排程,以讓 mrtg 產生正確的流量資料:

```
# crontab job
*/5 * * * * /usr/local/bin/mrtf /use/local/etc/mrtg.cfg
```

如果一切設定皆無問題，您即可以 Broswer 連到 mrtg 網頁查看，會有如下畫面(僅列出一張畫面)，而若您意欲於圖上顯示多條不同線形資料(如:Success/Recursion..等六類)，則您可參考 rrdtools 之用法，於 mrtg 的網站上可以找到資料或相關連結。



至於若您有兩部以上的 DNS, 而想將圖畫在一起，這個部份您可使用 ssh（直接執行遠端的主機的程式)，syslog(將 syslog 集中在一起分析)，或以 snmp 自訂 OID 的方式運形皆可(方法有許多種)。

第五章 DNS 與其他服務之結合

DNS 除一般我們所知道的，提供名稱解析之服務外，目前較受注意的尚有廣告信黑名單系統(DNSRBL，DNS Realtime Block List)，這是 Mail Server 在使用的。此外尚有一較新的技術為 Enum (tElephone NUmber Mapping)，主要做為電話號碼與網路位址之對應。

第一節 DNSRBL 廣告信黑名單

黑名單為什麼會使用到 DNS 呢 ?主要及是充份的利用 DNS 特性，其查詢快速、分散式環境、Round Robin 機制都是黑名單系統的重要基石(其實多數類似目錄系統之服務皆可使用 DNS 來實現)。而多數的黑名單組織都是由網路上的團體所公益發起，他們亦沒有足夠的資金購買昂貴的設備，故 DNS 即便成為大家的第一選擇。

著名之數個黑名單系統： mail-abuse.org/dnsrbl.org/orbs.org/vix.com/ aupads.org，以上都可以從他們的網站查到相關的訊息。而目前台灣並沒有相關組織，僅由 ISP 業者自行處理廣告信問題。

我們即先從 sendmail 範例來看其如何做到黑名單系統:

```
#sendmail.mc 節錄
FEATURE('blacklist_recipients')dnl     #要使用黑名單系統
FEATURE(dnsbl)dnl                      #以 dnsbl 來做
FEATURE(dnsbl,'spam.dnsrbl.net')dnl    #要用 dnsrbl 的 RBL 系統
FEATURE('dnsbl','rsbl.aupads.org','"550 Mail from " $&{client_addr} " refused: spam
site. See http://www.aupads.org/cgi-bin/rsbl-lookup?"$&{client_addr}""')dnl
                                       #也要用 aupads.org 的系統
```

上述為 sendmail.mc 檔案，經過 m4 處理後理後會轉成 sendmail.cf 供 sendmail 使用

```
m4 sendmail.mc > sendmail.cf
#sendmail.cf 節錄
#DNS based IP address spam list blackholes.mail-abuse.org
R$* $: $&{client_addr}
R$-.$-.$-.$- $: <?> $(dnsbl $4.$3.$2.$1.blackholes.mail-abuse.org. $: OK $)
#來源 IP 若為 140.112.55.55 則查詢 55.55.112.140.blackholes.mail-abuse.org. 是否
有記錄存在(如果查到 A 記錄,即是 SPAM 的 IP)
R<?>OK $: OKSOFAR
R<?>$+<TMP> $: TMPOK
R<?>$+ $#error $@ 5.7.1 $: "550 Rejected: " $&{client_addr} " listed at
lackholes.mail-abuse.org"
#如果找到 A RR 則表示這個 IP 在黑名單檔中，這封信將被退
# DNS based IP address spam list spam.dnsrbl.net
R$* $: $&{client_addr}
R$-.$-.$-.$- $: <?> $(dnsbl $4.$3.$2.$1.spam.dnsrbl.net. $: OK $)
R<?>OK $: OKSOFAR
R<?>$+<TMP> $: TMPOK
R<?>$+ $#error $@ 5.7.1 $: "550 Rejected: " $&{client_addr} " listed at
spam.dnsrbl.net"
```

廣告信的諸多問題我們並不多做介紹，若您感興趣可到上述的一些網站上參考。TWNIC 自身亦曾被列於黑名單當中，原因在於我們的 auto-reply 郵件造成別人誤會，只要寫封信去

說明一下，很快即可以解除，也有的黑名單系統是直接鎖住一般時間，時間到了就解除，這種寫信去是沒有用的，其程序網站上也都有說明。

第二節 ENUM 電話服務

DNS 及 PSTN 都有一個標準可行，透過 RFC（ Request for Comment ）的標準制定，將兩者再結合在一起。RFC 2915，2916，3026 對其中有著許多的說明，簡要說明如下：

1.財團法人台灣網路資訊中心電話 +886-2-23411313
2.去除 '-' '+' 後得 886223411313
3.將號碼反轉為 313114322688
4.轉成 DNS NAPTR (Name Authority PointER)記錄並加上 e164.arpa（ BIND 8.2.2 及 Windows 2000 SP1 版本後支援 NAPTR 資源記錄型式）

```
zone "6.8.8.e164.arpa" {
    type master;
    file "e164.arpa";};
```

5.在轄區檔案（ zone file ）中加入電話號碼的資源記錄

```
;IN  NAPTR  優先權   順序  旗標      正規表示法+URI                    取代項目
$ORIGIN 3.1.3.1.1.4.3.2.2.6.8.8.e164.arpa.
IN   NAPTR   100      10   "u"  "sip+E2U"     "!^.*$!sip:info@tele2.se!"      .
IN   NAPTR   102      10   "u"  "mailto+E2U"  "!^.*$!mailto:info@tele2.se!"
IN   NAPTR   104      10   "u"  "tel+E2U"     "!^.*$!tel:+886223413300!"
```

有
關相關欄位設定說明可參考 RFC2916 之說明，綜合以上表示:

1.若有人打電話給 +886-2-23411313，resolver 會先查詢到此號碼有三筆 NAPTR RR
2.比較優先權及順序後先使用 SIP(Session Initiation Protocol）服務，並轉成 E2U（ E.164 to URI）的型式連接。
3.若此一服務請求不成立，則使用 SMTP 服務來傳送語音郵件。
4.若再不行則使用轉接，它則會再做一次 Enum 的轉接服務，嘗試將電話轉至 +886-2-23413300，此時判斷有 tel:+ 時，會再做一次 Enum 的解析服務，服務內容則根據其電話號碼的 DNS 資源記錄來定義，若沒有相關記錄則使用類似一般傳統的轉接服務。

將電話號碼加入到 DNS 的記錄中後，結合 PSTN 和 IP 網路對映的關係，我們以下圖做實際的介紹：

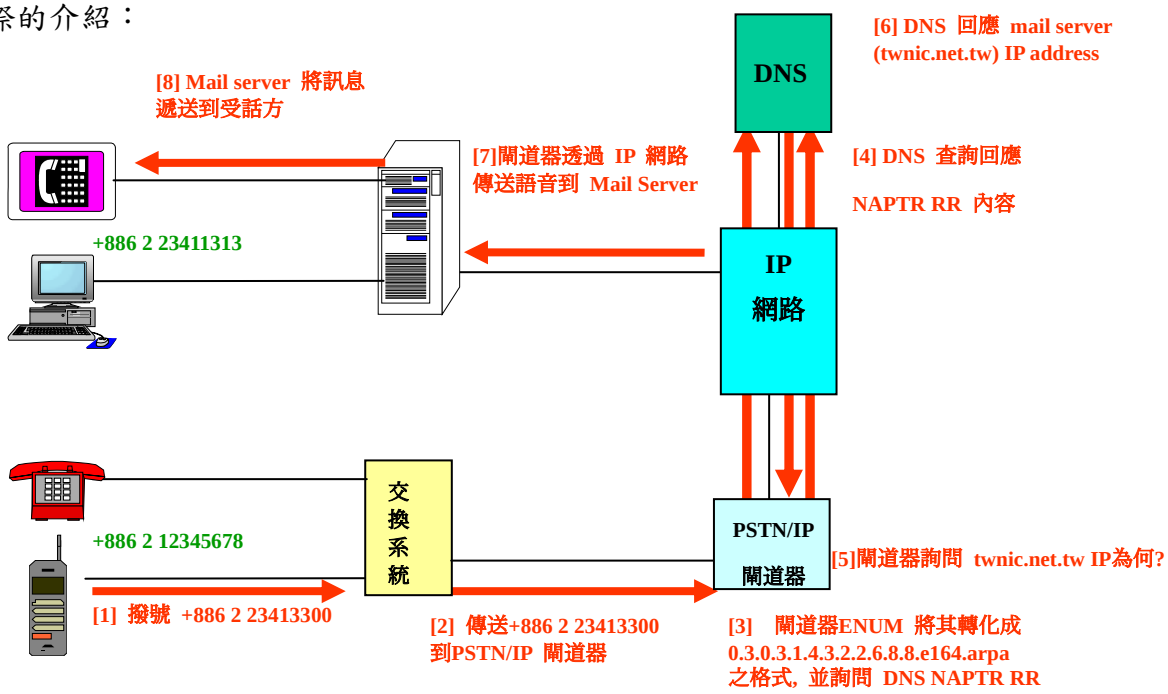


圖 ENUM PSTN 撥號至 IP 網路之流程

由以上我們可知 Enum 的 PSTN 及 IP 間的轉換重點在於 DNS 及「閘道器」(Gateway) 之間的協調，而這兩樣目前已都可以支援 Enum 的服務要求。

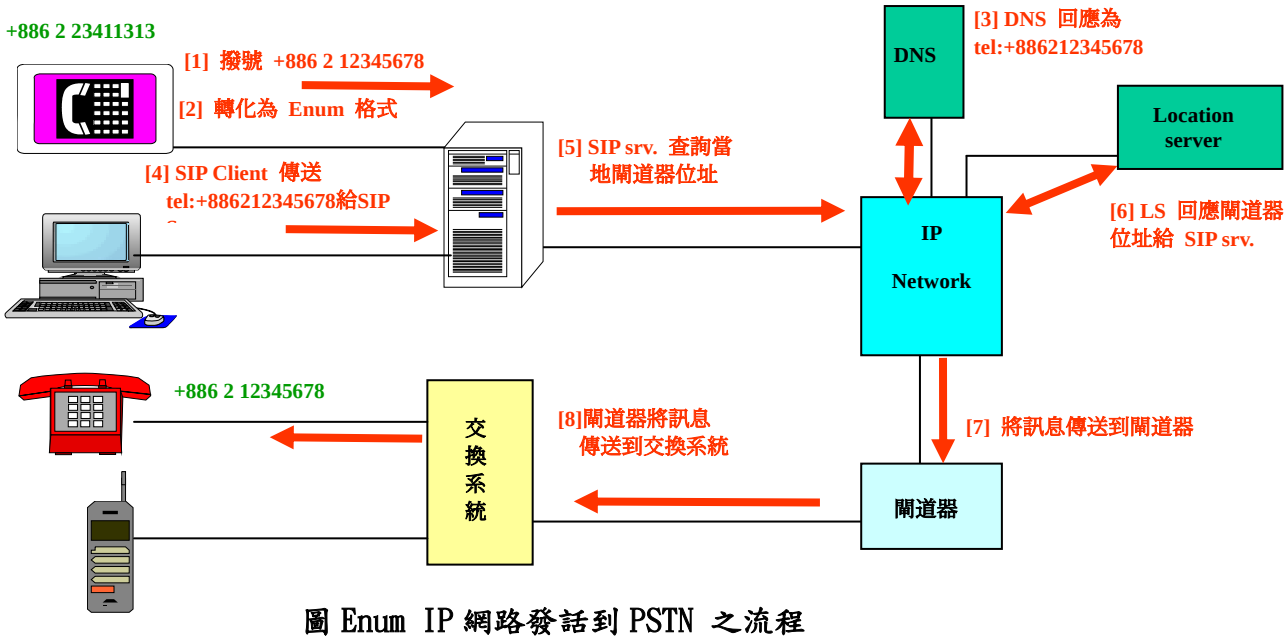


圖 Enum IP 網路發話到 PSTN 之流程

第六章 DNS 與網路安全

第一節 新聞報導

DNS 的安全性在整體的網路安全中佔有相當重要的地位，而卻常常被忽略，這除了因為不熟悉外，主要針對 DNS 的攻擊並不多見，但其影響卻超出一般人想像。

以下引用幾條新聞訊息:

http://www.sinica.edu.tw/info/security/news-%C0b%AB%C8%A7%F0%C0%BB%C2%EA%A9w.info%BA%F4%B0%EC%A8t%B2%CE.htm

駭客攻擊鎖定.info 網域系統

上周網域名管理公司 UltraDNS 遭一波網際網路垃圾資訊轟炸，造成掛在.info 和其他網域名稱上的網站伺服器癱瘓，管理員為維護系統正常運作而疲於奔命。

UltraDNS 執行長 Ben Petro 說，上周四（11 月 21 日）早晨，在網路使用量最大的四小時內，這波攻擊每秒鐘向連上網際網路的各個裝置發出將近 200 萬則查詢訊息，是正常量的數倍之多，導致 UltraDNS 的系統難以消受。

「這是我們經歷過規模最大的一次攻擊，」Petro 說。他強調，這場攻擊並未影響到該公司核心的網域名系統（domain name system，簡稱 DNS）服務，但系統管理員必須迅速行動，請提供 UltraDNS 網路連線的骨幹網際網路公司攔阻那些垃圾訊息。「從網路管理的觀點來看，這的確讓我們緊張萬分，」他說。

將近一個月前，也出現類似鎖定 DNS 根伺服器的攻擊。所謂 DNS 根伺服器（DNS root servers）指的是一種資料庫，內含電腦維持頂層網域所需的重要資料。這些網域的作用如同網際網路版的電話簿白頁（white pages），將網域名稱（例如 www.cnet.com）與數據形式的網際網路位址逐一配對。

Petro 說，有關當局可能正著手調查此案。

但調查人員要找出發動攻擊者的位置，談何容易。用垃圾訊息灌爆網路，也就是眾所周知的分散式阻斷服務攻擊（distributed denial-of-service attacks），通常都是駭客用偽造的來源位址、透過事前即侵入的伺服器所發動。運用這種雙重的間接手法，使得元兇難以追查。

但 Petro 說，揪出攻擊者的重要性與日俱增，因為最近網際網路攻擊的趨勢已改變，已從原先鎖定零星的公司，轉變成瞄準網際網路基礎設施本身發動攻擊。

「當你癱瘓亞馬遜網站時，傷害的只是亞馬遜，」他說。「但是當你擊垮.com、.org 和 .net 時，影響的是一國的國內生產毛額（GDP），受害的會是全國的經濟。」

UltraDNS 是網際網路協會（Internet Society）的會員，是.org 網域名的主要 DNS 供應者。UltraDNS 另外也提供 .info 網域，以及愛爾蘭、盧森堡和挪威的頂層網域及其他九種網域。

「現實是，網路攻擊規模愈來愈大、愈演愈烈而且愈來愈快速，」Petro 說。「如同恐怖攻擊，你不知攻擊行動何時發生、如何發生。除非我們密切關注這些攻擊事件，而且追查出源頭，我們都可能淪為受害者。」

(http://www.secretchina.com/news/articles/2/10/23/27110b.html)

全球互聯網系統核心 21 日遭規模最大攻擊

2002 年 10 月 23 日 星期三

美國官員及電腦專家表示，掌管全球網際網路交通的十三座根名稱伺服器（root server）二十一日晚間曾同時遭不明人士攻擊達一小時，目的顯然是要癱瘓全球網路，這是網際網路創立以來所遭受規模最大、也最複雜的網路攻擊行動。

據自由時報綜合二十二日外電報導，美國加州「網路軟體協會」負責人維克西表示，上述攻擊行動是在美國東岸時間二十一日下午五時（台北二十二日清晨五時）左右開始，歷時約一小時，網際網路領域名稱系統（ＤＮＳ）的十三座根名稱伺服器同時遭人以「分散式阻斷服務」方式攻擊，造成部份網路交通受阻，但一般網路使用者並未察覺有異。

一位電腦專家指出，由於二十二日下午數個網站也曾遭到類似的攻擊，他猜測，這很可能是同一票人尋找新目標攻擊作樂。

美國聯邦調查局（ＦＢＩ）發言人指出，該局轄下的全國基礎建設保護中心已得知此事，並展開調查行動。美國官員也證實，十三座根名稱伺服器中，有九座在攻擊期間無法正常運作，但在電腦專家立即採取防禦措施，以及攻擊歹徒忽然停手後，網路交通隨即恢復正常。

參與搶救的匿名電腦專家還透露,他們曾與白宮國土安全辦公室和布希總統的關鍵基礎建設保護署合作。

網路領域名稱系統可讓各電腦網路系統在使用者鍵入的文字名稱以及用數字代碼來表示的網址之間進行轉換，但仍必須仰賴根伺服器來提供網址資訊，因此這十三座分布在全球各地，由美國政府機關、各大學、企業和私人組織負責運作的根伺服器一直都被網路安全專家視為網際網路最大的弱點，很有可能在網路遭人惡意攻擊時當機。

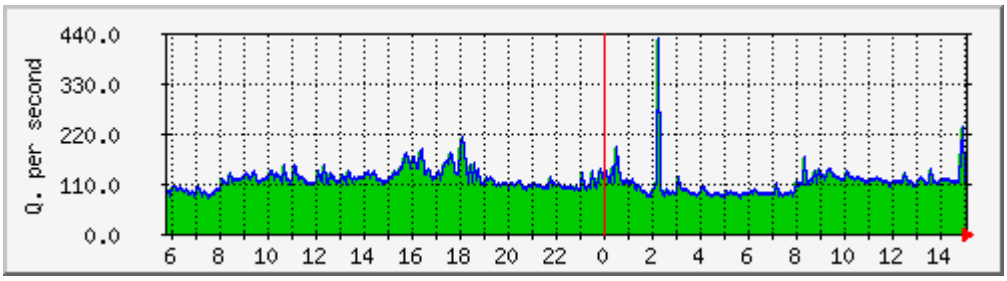
據維克西指出，網路根伺服器過去也曾遭到攻擊，但十三座伺服器同時遭到攻擊則相當罕見,但此次攻擊行動也證明瞭網路無法輕易被阻斷,因為網際網路設計的原旨就是要繞過障礙。

其他電腦專家則表示，此次攻擊未對網路造成重大影響，主因是許多網路服務供應者及大型企業或組織都會儲存或「快取」最受歡迎的網路資訊分類目錄，而不必全部仰賴根伺服器進行轉換的緣故。雖然理論上網路可在僅有一座根伺服器的情況下運作，但假若四座以上的根伺服器同時當機時間過長，網路運作的速度就會明顯減慢.

上述二篇轉載文章我們可以看出 DNS 安全的重要性，而愈處上層就需更考慮到安全性，如果 .tw 的 Root Server 不安全，那全台灣的 DNS 可能都會發生問題，即便是全台的網路會出問題。而若 ISP 的 DNS 不安全，因其提供眾多的用戶查詢，亦可能產生重大的問題。一般公司行號的 DNS 雖然不會造成如此大事件，但是依然可能對公司造大重大傷害，故安全性問題不可不慎。BIND 安全與版本相關，詳細版本與系統漏洞對應關係請參考(若您在列見議您更新版本較好)

http://www.isc.org/products/BIND/bind-security.html

若您使用 mrtg 做監測時，可使用 mrtg 之 Threshold 之功能，做到異常時通知之功能，您可以參考這篇文章之介紹:
http://redhat.ecenter.idv.tw/bbs/showthread.php?threadid=43276
如，當下圖的流量超過 400 次/秒 查詢時，就可以送信到管理者，讓其了解，早作因應。



第二節 版本昇級、Patch

若您打算從 BIND 8.x 版本昇級到 9.x 的版本，您的設定檔（named.conf）及 Zone File 皆能不改變而相容於 9.X 的版本，所以您毋需擔憂昇級所帶來的相容性問題。

程式或系統難免都會有漏洞存在，當漏洞被發現時，系統管理人員基本上皆應於第一時間做出反應以免遭受到危害，這反應可能是隔離，或是修補系統。隔離後依然是要修補系統，最重要的即是建構一個安全的環境，故系統昇級或更新的之問題不可不知。

昇級

昇級是放棄現行使用的版本，改換到新的版本，即是重新安裝新的 BIND 程式

```
[root@pc071 SIP]# tar -zxvf bind-9.2.2.tar.gz    #以 tarball 方式安裝系統
[root@pc071 SIP]# cd bind-9.2.2                  #進入 BIND 9.2.2 之目錄
[root@pc071 bind-9.2.2]# ./configure -help       #查看安裝設定功能項目
                                                 #此處僅介較重要的項目
--prefix=PREFIX                                  #BIND 安裝路徑，預設為 /usr/local
--enable-ipv6                                    #是否支援 IPv6，預設為自動偵測網路環境
--enable-threads                                 #是否支援 multi-thread 架構
--with-openssl=DIR                               #是否支援 SSL ( for DNSSEC)
```

[root@pc071 bind-9.2.2]# ./configure --prefix=/usr/local --enable-ipv6 --enable-pthread
….執行畫面略
[root@pc071 bind-9.2.2]#make
[root@pc071 bind-9.2.2]#make install

安裝時請記得先移除您先前之版本，以免兩者路徑不同時，同時兩種版本存在系統上造成混淆，也避免日後不小心啟動到舊的版本，如果您使用 RPM 的方式安裝系統，這方面的檢查即會方便許多(幫您移除舊並安裝新版)

[root@pc071 bind-9.2.2]#rpm -Uvh -force -nodeps new-version-of-bind.rpm

Patch

以修補的方式來更新系統其實和上面差不多，但可能因為現行的版本不方使更動下方以修補方式進行，修補檔常以 .diff 或 .patch 存在，且會存在其版本名以供判斷:

bind-9.2.0.patch *# 9.2.0 版本的修補*
bind-9.2.0-9.2.2.patch *# 9.2.0 昇級到 9.2.2 之修補*
bind-8.2.3-8.3.3.patch *#8.2.3 昇級到 8.3.3 之修補*

修補檔只是兩個版本間的**差異點**，而修補檔以 diff 來產生:

diff -ur /root/bind-9.2.0/bin/check/named-checkzone.c /root/bind-9.2.2/bin/check/named-checkzone.
--- /root/bind-9.2.0/bin/check/named-checkzone.c *# 三個 - 符號，表示原始檔*
+++ /root/bind-9.2.2/bin/check/named-checkzone.c *#三個 + 符號，表示目的檔*
@@ -15,7 +15,7 @@ *#原始檔的第 15 行後的 7 行內容*
** WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.*
**/*

- / $Id: named-checkzone.c,v 1.13.2.2 2002/02/08 03:56:58 marka Exp $ */*
+ / $Id: named-checkzone.c,v 1.13.2.3 2002/07/11 05:44:10 marka Exp $ */*
 # - 表示換掉這行，改成 + 號這一行
#include <config.h> *#以下略*

有了修補檔，我們可以對 9.2.0 的版本進行昇級到 9.2.2 ，從上述的例子我們要注意的是路徑的問題，如果我們現在目錄位於 9.2.0 中，此時我們需略過 /root 的路徑，則需以下列命令執行:

[root@pc071 bind-9.2.0]# patch -p1 < bind-9.2.0-9.2.2.patch *#-p1 表示略過目錄的第一層*
patching file ./bin/check/named-checkconf.html *# patch 了那些檔案*
patching file. /bin/check/named-checkzone.8
patching file. /bin/check/named-checkzone.c

修補好後再以上一章節之 configure;make;make install 方式安裝即可完成系統。

第三節 SPOF(Single Point of Failure)之問題

SPOF 即是只有一部 DNS 之問題，這種狀況其實不符合 InterNIC 的規定，以 TWNIC 而言，做 DNS 指定時，都要您填寫兩部以上之資料，其原因:

容錯：若您僅有一部 DNS 主機時，相信很多人都有經驗的是該主機失效後，網路的眾多功能也會跟著失效(Web/Mail…)，但若您有兩部以上的 DNS 主機則大大降低了這樣的可能性。若您的 DNS 分佈在不同的 ISP 網段，更可降低網路斷線所造成的眾多問題。

負載平衡：若您設定了兩部以上的 DNS 主機，當有人連接您的網站前，其接受 DNS 查詢乃是兩台主機輪流運作(輪詢，Round-Robin)。在這樣的運作機制下讓您的系統可以更穩定。

系統安全：目前網路上有著許多可以針對 DNS 作攻擊的程式，若您擁有兩台以上的主機，可降低許多來自於攻擊的危險(至少增加了一倍的安全性，其中不同的主機尚可運作不同的作業系統)。因為絕大多數的 DNS 查詢或攻擊皆是使用 UDP 協定(User Datagram Protocol)，在攻擊發生時較不容易被查覺。

SPOF 問題最有名案例即是 2001 年時，微軟的四部 DNS 主機皆擺在同一個 LAN 之下，而當這個 LAN 對外的 Router 被攻垮後，所有的 DNS 即失去作用。進而，對許多人而言，微軟好像從網路上消失一般。所以，若您的單位對安全議題較重視，不能忽略失去 DNS 對您網路的影響。

第四節 遠端溢位問題/拒絕服務存取

遠端溢位(Remote buffer overflow) 是多數系統不能避免的安全漏洞，而 BIND 某些版本亦存在此一問題,您可參考 http://www.securityfocus.com/cgi-bin/sfonline/vulns.pl 網址，於 Vendor 處選擇 ISC ，則可列出 BIND 所有的系統問題，其中我們選擇 TSIG Vuln，這是一個可以遠端溢位的漏洞，於 http://www.securityfocus.com/data/vulnerabilities/exploits/tsig.c 可以取得溢位的範例程式。由 tsig 的漏洞公告中，我們知道這個問題可能被取得系統權限，而受影響的版本幾乎包括了 8.2.3(不含)以下所有的版本。由此可見,任何人其實都可以很容易取得這一類資源。其中最有名的即是當年度之 Lion Worm，以類似 Code Red 型式之 Worm 在網路上尋找 DNS 主機並攻擊入侵，最後將 /etc/passwd /etc/shadow 檔寄給作者。

拒絕服務存取的攻擊目前已不可勝數,其目的多不在於入侵系統,而是癱瘓整個網路系統，這個問題目前幾乎是無解，僅能以被動的方式來防禦，同樣的在 http://www.securityfocus.com/ cgi-bin/sfonline/vulns.pl 的資料上您也可以看到許多這類的資訊。

基本上亦建議，若環境允許，您不要將所有的的服務皆擺在同一部主機，因為一旦某一服務發生問題(如 Mail/Web/DNS/…)，其他服務可能皆受影響，對您單位的網路運作影響甚大。

上述兩種網安的狀況屢見不鮮，不過這類攻擊因為特徵明顯，多少可由入侵偵測系統(IDS)或防火牆所發現，但 DNS 欺騙則可能較不容易被偵測到。

第五節 DNS 欺騙之問題

DNS 欺騙之案例並不多見，但不代表較少，而是其方式較難查覺。所謂的欺騙即是告訴您假的資料，如果您欺騙 ISP 的 DNS ，告訴他 www.kimo.com.tw 在 1.2.3.4，在某段時間內，該 DNS 當有人查詢 www.kimo.com.tw 時，它會回應 1.2.3.4 的位址。當然一般人使用 kimo 可能只是單純的瀏覽，但同樣的欺騙的 FQDN 可以是銀行或具有權益的位址，而不法之人再以其他手法套取您的帳戶資料。

欺騙手法一

我們先看看原來 DNS 查詢所顯示的內容

```
[root@pc071 named]# dig a www.twnic.com.tw. //查詢 www.twnic.com.tw. 的 A RR
; <<>> DiG 9.2.1 <<>> a www.twnic.com.tw.
;; global options:  printcmd
;; Got answer:       //DNS Packet 共分五段 (RFC
1035),Header/Query/Answer/Auth/Additional
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 21970       //這裏會顯示
DNS Packet Header
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 3, ADDITIONAL: 3
// 知道查詢 id, DNS Packet 類別為查詢(Query),回答的答案(ANSWER)1 筆
// 權威主機三筆(即該域名之 NS RR) 及 ADDITIONAL 3 筆 ( NS glues A RR)
;; QUESTION SECTION:     //query 的 RDATA 及 CLASS/TYPE
;www.twnic.com.tw.                       IN      A

;; ANSWER SECTION:       //回答的結果
www.twnic.com.tw.         86400   IN      A       211.72.211.71

;; AUTHORITY SECTION:    // 目前的結果都正確無誤
twnic.com.tw.             86400   IN      NS      dns.hinet.net.
twnic.com.tw.             86400   IN      NS      ns1.twnic.com.tw.
twnic.com.tw.             86400   IN      NS      ns2.twnic.com.tw.

;; ADDITIONAL SECTION:
dns.hinet.net.          86400   IN      A       168.95.192.1
ns1.twnic.com.tw.         86400   IN      A       211.72.211.1
ns2.twnic.com.tw.         86400   IN      A       211.72.211.2

;; Query time: 3 msec
;; SERVER: 211.72.211.71#53(211.72.211.71)
;; WHEN: Tue Apr 22 10:24:01 2003
;; MSG SIZE  rcvd: 159
```

這個結果是對的範例，我們查詢任一 FQDN 基本上皆會以類似這個結果呈現，如果這是一個合法的域名（指經向 TWNIC 申請過），那別人無論向任何一台機器查詢都會得到此一結果，唯一的差別可能是 TTL 值不同（因為 TTL 值會倒數）。

現在我們在這台機器上加上一個 hinet.net 的域名，並設定其 Zone File:

```
zone "hinet.net" {
        type master;
        file "hinet.net.hosts";
};
```

```
;Zone File
$TTL    86400
@       IN      SOA ns1     root (2002021301 1D 1H 1W 2D )
$ORIGIN hinet.net.
        IN      NS      dns.hinet.net.
dns     IN      A       211.72.211.71
```

修改後我們再查詢一次同樣的 FQDN

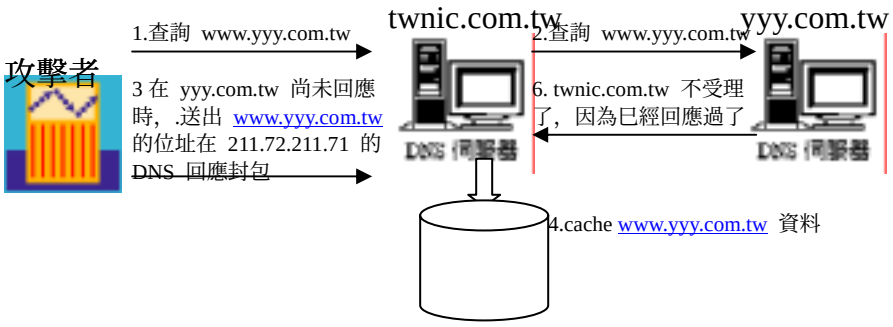
```
[root@pc071 named]# dig a www.twnic.com.tw.

; <<>> DiG 9.2.1 <<>> a www.twnic.com.tw.
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 46120
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 3, ADDITIONAL: 3

;; QUESTION SECTION:
;www.twnic.com.tw.                      IN      A

;; ANSWER SECTION:
www.twnic.com.tw.       86400   IN      A       211.72.211.71

;; AUTHORITY SECTION:
twnic.com.tw.           86400   IN      NS      dns.hinet.net.
twnic.com.tw.           86400   IN      NS      ns1.twnic.com.tw.
twnic.com.tw.           86400   IN      NS      ns2.twnic.com.tw.

;; ADDITIONAL SECTION:
dns.hinet.net.          86400   IN      A       211.72.211.71
ns1.twnic.com.tw.       86400   IN      A       211.72.211.1
ns2.twnic.com.tw.       86400   IN      A       211.72.211.2

;; Query time: 3 msec
;; SERVER: 211.72.211.71#53(211.72.211.71)
;; WHEN: Tue Apr 22 10:40:10 2003
;; MSG SIZE  rcvd: 159
```

這個時候，我們可以發現 dns.hinet.net 的位址被我們換到 211.72.211.71，如果別部 DNS 查詢 www.twnic.com.tw 時，它同時會 Cache Auth/Additional 的資料，也就 Cache 了 dns.hinet.net IN A 211.72.211.71，如果您技術高明些，寫一個程式去找 Internet 的 DNS Server，然後叫他查 www.twnic.com.tw.，它很可能就會 Cache dns.hinet.net 在你家了，當然你亦可以用 www 或 mail 等較有義意的資料來欺騙。

這個問題或許您會意外於它的簡單，但在 BIND 8.1.2 以前的版本及 Windows 上(Windows 作者並不清楚版本狀況，但至少確認沒上過 Service Pack 一定是可以 Spoofing 的)却普遍存在，或許您同時亦會想應如何避免被欺騙呢(請參閱附錄或 http://www.acmebw.com/resources/papers/securing.pdf) ?

欺騙手法二

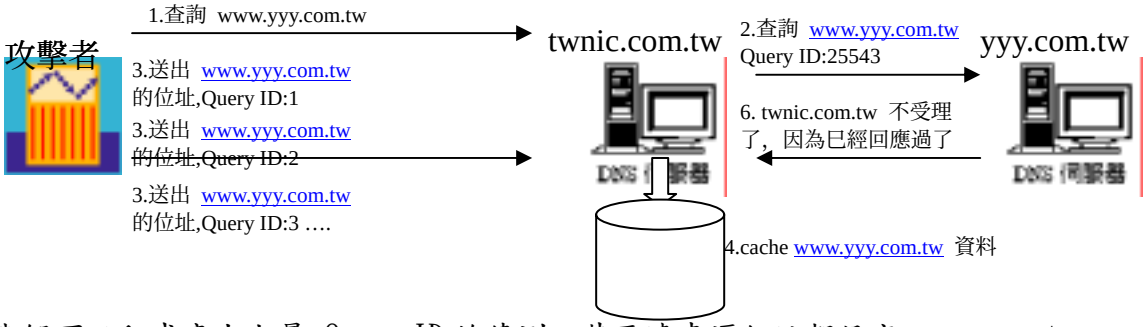
手法二不需以 DNS Server 做為媒介，直接以封包的方式欺騙 DNS



欺騙手法三

手法一二的方法過於簡單，故後來 ISC 即加強此一問題，也就是每原來每個查詢動作皆會保留其 query id 以供識別，不過依據 DNS Packet 特性，Query ID 為 2 bytes，故在數值範圍有限之下(0~65535)，這種問題還是有可能會發生的，尤其是數個版本的 BIND 每次的 Query ID 以加一方式遞增。後來雖改成 Random 方式產生，但系統預設 (use-id-pool)並不會保留每次 query 的位址 與 query ID 的對應。所以，若對 DNS 的設定或運作不甚了解的話，問題還是依舊。

即使你使用了 use-id-pool 做 Query ID 的記錄，還是有可能以大量的封包所猜中



我們可以程式產生大量 Query ID 的猜測，甚至連來源位址都設定 yyy.com.tw. ，雖然可能還未猜到 Query ID，正確的 DNS 封包已經回應了，不過做個 N 次總會猜中的，而只要猜中一次對攻擊者而言就已經足夠了，因為他可以將 TTL 值設的相當高。

欺騙手法四

手法四即是以一般的遠程溢位方式，入侵後修改 DNS 資料，將 TTL 值設定最大，然後以程式令外界的 DNS 向其查詢，使用 Cache 此一有問題的資料，即使用日後系統管理人員發現後，要令 DNS 回復正常仍需一段時日，通常這種做法較少人用，但多用於截收 Email 或網站轉向。

任何一種 DNS 欺騙並不限於 A RR，即使反解亦是可以是欺騙的對象，因為反解涉及許多服務的認證，要避免自己成為被欺騙的對像，除了注意版本外，設定要完整，並檢測記錄，或是可以以排程方式定期重啟 DNS 服務亦可。

第六節 不安全的 DNS 對企業網路的影響

DNS 失常

這是最常見的情況，使用者會感覺到 DNS 失去作用。此時除了重新啟動，還需去了解為什麼 DNS 會失效。

假造網頁

透過中介（ man in the middle ）的手法，很容易的騙出使用者名和密碼，或者其他敏感資料。其方法如原來您的 Web 主機在 IP1，駭客 DNS 入侵成功後將 Web 指向 IP2，此時使用者以 www.yourdomain.com.tw 來連線時，將指向 IP2，駭客再以映射手法(mirror)讓使用者覺得網頁是正常的，造成使用者的錯覺。此時即可能騙出使用者登錄時的重要資訊，這種狀況下將損及使用者權益及公司信譽。之前國內曾發生過多起這樣的案例。

複製郵件

所有的信件到達你的服務器之前可以被拷貝，修改或者刪除。入侵者只要了解郵件伺服器與 DNS 的運作原理輕易即可達成此一目的，而其也可以偽造成您的信件寄出，這些都是可以透過 DNS 完成，而您不會感覺到很明顯的異常。其手法同上一段所述。

授權問題

某些與信任有關服務（如 mail,firewall,proxy 等等）若涉及 DNS 域名信任時將會無效。如您的防水牆信任 any.com.tw 網域可自由通過，在 DNS 被入侵後防火牆將完全失效。因為入侵者可在您的 DNS 中添加他機器為 any.com.tw 網域機器的資訊。

系統權限

因為部份的 DNS 是以系統管理者權限在運行，當駭客從 DNS 入侵後(指遠端溢位攻擊，remote buffer overflow ），通常亦直接取得系統權限，即使以非系統權限的身分執行，但依然會有潛在的風險。

資訊洩漏

DNS 設定不正確或不完整，可能造成資訊洩漏的問題，最常見的就是 Zone Transfer。

附錄一 BIND 8.X 的安全設定

作者：backend
主頁：http://www.nsfocus.com
日期：2001-3-09
---[[前言]]---

為什麼要寫這篇文章？第一個原因當然就是前段時間出現的 BIND 8.2.x TSIG 安全漏洞(還有去年公布的 BIND 8.1.x/8.2.x NXT 安全漏洞)，直到目前為止，國內也還沒有關於 DNS 服務安全配置方面的較為完整的文章(即使是國外也不多見)。另一個原因是經過調查發現，幾乎任何一種 UNIX 家族的操作系統，都使用 BIND 軟體
作為其 DNS 的唯一實現，比起其它諸如 ftp/http/pop3 等網絡服務有各種各樣的發行版本，所以一旦被發現有安全問題，則受影響的主電腦之多也是其它漏洞很難比擬的。所以覺得應該寫一份針對 BIND DNS 服務軟體的安全配置資料，充分利用 BIND 自身已經實現的保護功能，加強 BIND 安全性，從而能抵御目前已知的 BIND 安全漏洞，並使潛在的安全漏洞所可能對服務器造成的影響盡可能地減少。

配置環境：
FreeBSD 4.1-RELEASE
BIND 8.2.3
---[[啟動安全選項]]---

named 進程啟動選項：

-r：關閉域名服務器的遞歸查詢功能（缺省為打開)。該選項可在配置文件的
options 中使用"recursion"選項覆蓋。

-u 和-g ：定義域名服務器運行時所使用的 UID 和 GID。
這用於丟棄啟動時所需要的 root 特權。

-t ：指定當服務器進程處理完命令行參數后所要 chroot()的目錄。

---[[配置文件中的安全選項]]---

1、假如希望記錄安全事件到文件中，但同時還希望保持原有的日志模式，可以添加以下內容：

```
logging {
channel my_security_channel {
file "my_security_file.log" versions 3 size 20m;
severity info;
};
category security {
my_security_channel;
default_syslog; default_debug; };
}
```

其中 my_security_channel 是使用者自定義的 channel 名字,my_security_file.log 是安

全事件日志文件，可包含全路徑（否則是以 named 進程工作目錄為當前目錄）。安全事件日志文件名為 my_security_file.log，保存三個最近的備份（my_security_file.log0、my_security_file.log1、my_security_file.log2），日志文件的最大容量為 20MB（如果達到或超這一數值，直到該文件被再次打開前，將不再記錄任何日志消息。缺省（省略）時是沒有大小限制。）

2、在 options 節中增加自定義的 BIND 版本資訊，可隱藏 BIND 服務器的真正版本號。
version "Who knows?";
// version 9.9.9;
此時如果通過 DNS 服務查詢 BIND 版本號時，返回的資訊就是"Who knows?"。^_^

3、要禁止 DNS 域名遞歸查詢，在 options（或特定的 zone 區域）節中增加：
recursion no;
fetch-glue no;

4、要增加出站查詢請求的 ID 值的隨機性，在 options 節中增加：
use-id-pool yes;
則服務器將跟蹤其出站查詢 ID 值以避免出現重復，並增加隨機性。注意這將會使服務器多占用超過 128KB 記憶體。(缺省值為 no)

5、要限制對 DNS 服務器進行域名查詢的主電腦，在 options（或特定的 zone 區域）節中增加：
allow-query { };
address_match_list 是允許進行域名查詢的主電腦 IP 列表，如"1.2.3.4; 5.6.7/24;"。

6、要限制對 DNS 服務器進行域名遞歸查詢的主電腦，在 options（或特定的 zone 區域）節中增加：
allow-recursion { };
address_match_list 是允許進行域名遞歸查詢的主電腦 IP 列表，如
"1.2.3.4; 5.6.7/24;"。

7、要指定允許哪些主電腦向本 DNS 服務器提交動態 DNS 更新，在 options（或特定的 zone 區域）節中增加：
allow-update { };
address_match_list 是允許向本 DNS 服務器提交動態 DNS 更新的主電腦 IP 列表，如
"1.2.3.4; 5.6.7/24;"。
缺省時為拒絕所有主電腦的提交。
8、要限制對 DNS 服務器進行區域記錄傳輸的主電腦，在 options（或特定的 zone 區域）節中增加：
allow-transfer { };
address_match_list 是允許進行區域記錄傳輸的主電腦 IP 列表，如"1.2.3.4; 5.6.7/24;"。

9、要指定不接受哪些服務器的區域記錄傳輸請求，在 options（或特定的 zone 區域 ）節中增加：
blackhole { };

address_match_list 是不接受區域記錄傳輸請求的主電腦 IP 列表，如"1.2.3.4;
5.6.7/24;"。

10、在 options 節中還有一些資源限制選項，不同使用者可根據實際情況靈活設置，但一定要注意不當的設置會損失 DNS 服務的性能。
coresize ; // core dump 的最大值。缺省為 default。
datasize ; // 服務器所使用的最大資料段記憶體。缺省為 default。
files ; // 服務器能同時打開的最大文件數。缺省為 unlimited（不限制）。注意，並非所有操作系統都支持這一選項。）
max-ixfr-log-size ; // （目前版本暫不使用。）限制增量區域記錄傳輸時會話日志的大小。
stacksize ; // 服務器所使用的最大堆棧段記憶體。缺省為 default。

11、定義 ACL 地址名（即用於上面的）。注意，如果要使用
這里定義的列表名，必須先定義，后使用！ 例如：
acl intranet {
192.168/16;
};
acl partner {
!172.16.0.1;
172.16/12; // 除 172.168.0.1 外 172.16.0.0/12 網絡中其它主電腦
};

BIND 已內置以下四個 ACL：
all // 允許所有主電腦
none // 禁止所有主電腦
localhost // 本機的所有網絡接口
localnets // 本機所在網絡

12、BIND 域名服務器的一個有用功能（慎用！！！）：
控制管理接口 controls 節語法格式：
controls {
[inet ip_addr
port ip_port
allow { ; };]
[unix path_name
perm number
owner number
group number;]
};

controls 節提供管理接口。如果使用第一種(inet)，則在指定 IP（接口）和端口上監聽，但只允許在 allow 中限定允許與其連接的 IP 地址列表。如果使用第二種 (unix)，則產生一個 FIFO 的控制管道，權限、屬主和使用者組都由其參數限定。

---[[通過 TSIG 對區域記錄傳輸進行認證和校驗]]---------------------------
首先請確保你的 BIND 域名服務器軟體已更新到最新版本！ 在 BIND 8.2+中，能夠使用事務簽名（Transaction Signatures，即 TSIG！） 來對區域記錄資料進行驗證和校驗。它要求在主域名服務器和輔助域名服務器上配
置好加密密鑰，並通知服務器使用該密鑰與其它域名服務器通信。(注意，TSIG 的使用要求域名服務器必須進行時鐘同步！)

A、如果需要用 TSIG 簽名來進行安全的 DNS 資料庫手工更新，具體操作步驟很簡單：

1、使用 BIND 自帶的 dnskeygen 工具生成 TSIG 密鑰。
dnskeygen -H 128 -h -n tsig-key.
則會生成兩個文件。'Ktsig-key.+157+00000.key'內容如下：
tsig-key. IN KEY 513 3 157 awwLOtRfpGE+rRKF2+DEiw==
'Kvip-key.+157+00000.private'內容如下：
Private-key-format: v1.2 Algorithm: 157 (HMAC) Key: awwLOtRfpGE+rRKF2+DEiw==
注意這些密鑰都已經過 BASE64 編碼了。將它們放到本地域名服務器的配置文件中。例如 ：
key tsig-key. { algorithm hmac-md5; secret "awwLOtRfpGE+rRKF2+DEiw=="; };
zone "dns.nsfocus.com" {
... ...
allow-update { key tsig-key. ; };
}
記住要重啟 named 守護進程。

然后將這兩個密鑰文件復制到客戶端系統（或輔助域名服務器），例如為/var
/named/tsig 目錄。最后運行如下命令即可：
nsupdate -k /var/named/tsig:tsig-key.

B、如果需要對區域記錄傳輸（自動或手工）進行 TSIG 簽名，則：
1、用 dnskeygen 生成 TSIG 密鑰，方法同上。

2、主域名服務器配置文件的內容（節選）如下：
// 定義認證的方法和共享密鑰
key master-slave {
algorithm hmac-md5;
secret "mZiMNOUYQPMNwsDzrX2ENw==";
};
// 定義輔助域名服務器的一些特性
server 192.168.8.18 {
transfer-format many-answers;
keys { master-slave; };
};
// 區域記錄定義
zone "nsfocus.com" {
type master;

```
file db.nsfocus.com;
allow-transfer { 192.168.8.18; };
};
```

3、輔助域名服務器配置文件的內容（節選）如下：

```
// 定義認證的方法和共享密鑰
key master-slave {
algorithm hmac-md5;
secret "mZiMNOUYQPMNwsDzrX2ENw==";
};
// 定義與主域名服務器通信時的一些特性
server 192.168.8.19 {
transfer-format many-answers;
keys { master-slave; };
};
// 區域記錄定義
zone "nsfocus.com" {
type slave;
file "bak.db.nsfocus.com";
masters { 192.168.8.19; };
allow-transfer { none; };
};
```

---[[實施 DNSSec 功能]]----------

說實在的,我一直在考慮需不需要在目前的版本中實施 DNSSec 功能。因為雖然 ISC 早已在 BIND 8.1.x 版本后增加了 DNSSec 的實現，但實際的應用卻不常見，而且去年公布的 NXT 遠程安全漏洞和 DNSSec 有關（實際上 NXT 就屬於 DNSSec 實現的功能之一）。最后我決定在本文不討論如何實施 DNSSec 安全功能。但不可否認，DNSSec 確實是一項很好的安全技術，它通過加密 DNS 資料來提高了 DNS 服務的安全性。主加密密鑰用於對第一級域名的加密密鑰進行加密簽字，第一級域名(.com, .cn 等)密鑰用於對自身資料及其子域名密鑰進行加密簽名，以此類推。
例如，nsfocus.com 的域名服務器由.com 域密鑰簽名，nsfocus.com 域密鑰則用於對 www.nsfocus.com 域名進行加密簽名。

---[[實現 BIND 的 chroot]]---------

（以 FreeBSD 系統平台為例）
步驟一：BIND-8 的最新源代碼版本獲取和安裝

請到 ISC FTP 站台下載 BIND 的最新版本。
BIND 8：http://www.isc.org/products/BIND/bind8.html
BIND 9：http://www.isc.org/products/BIND/bind9.html

步驟二：構造靜態(static)的 named 和 named-xfer 二進制文件

在編譯和安裝后，你需要構造可執行文件的靜態連結版本。只要對%BIND%/src /port/freebsd

目錄下的 Makefile.set 文件稍加修改后即可。
　　修改文件內容：
'CDEBUG= -O2 -g'
替換為：
'CDEBUG= -O2 -static'

切換到 BIND 的源代碼路徑，執行"make clean"和"make"命令。在下面的步驟中將會把這些文件復制到 chroot()目錄下。
```
# cd /tmp/bind/src
# make clean ; make
```

本步驟構造的靜態連結執行文件在運行時無需裝載動態連結庫。在 chroot()環境中，這種“獨立”可執行文件可避免出現缺少連結庫文件問題。它在 chroot()環境中無需任何靜態連結庫，可使服務配置簡單化。其它所有的網絡守護進程也可以編譯和使用這種靜態連結版本。

步驟三：構造 BIND 目錄
為 chroot()環境構造 BIND 目錄。這個目錄將在 chroot()環境中被 BIND 當作系統根目錄。在這里我使用/chroot/bind 作為 chroot 后的根目錄。
```
# cd /chroot/bind
# mkdir /chroot
# mkdir /chroot/dev
# mkdir /chroot/etc
# mkdir /chroot/etc/namedb
# mkdir /chroot/usr
# mkdir /chroot/usr/sbin
# mkdir /chroot/var
# mkdir /chroot/var/run
```

　　需要復制以下文件到其下的相應子目錄中，和進行一些必要的處理：
```
# cp /etc/namedb/named.conf /chroot/bind/etc/
# cp /etc/localtime /chroot/bind/etc/
# grep bind /etc/group > /chroot/bind/etc/group
# cp -R /etc/namedb/ /chroot/bind/etc/namedb/
# mknod /chroot/bind/dev/null c 2 2
# chmod 666 /chroot/bin/dev/null
# cp /tmp/bind/src/bin/named/named /chroot/bind/usr/sbin/
# cp /tmp/bind/src/bin/named-xfer/named-xfer /chroot/bind/
```

另外還可根據需要指定日志記錄目錄（如/var/log），請參考下面的章節或 named.conf 的手冊頁。

步驟四：添加 bind 使用者和組（如果沒有的話。如果已經有 bind 或 named 之類的使用者和組，請跳過本步驟。）
在/etc/passwd 和/etc/group 文件中添加 bind 使用者和組。它們是 DNS 服務器運行時的

UID/GID。 此時，你可以到 chroot 環境中執行
chown -R bind.bind /chroot/bind/etc/namedb
命令。這樣當你向系統送出中斷信號(kill -INT)時，named 進程能夠保存服務器緩存和統計資訊。如果該目錄為 root 所有則 named 進程無法將輸出寫到目錄中，但不會影響 named 服務器功能。另一個選擇是僅改變目錄權限（使 named 使用者具有寫權限），而屬主仍然是 root。這種方法也是可行的，但必須小心設置，確保其它使用者不會修改 named 記錄！

*** 重要警告***
不要用一個已存在的 UID/GID（如"nobody"）運行 named。記住，以 chroot 環境中使用任何已存在的 UID/GID 都可能會影響到服務的安全性。必須養成在 chroot 環境中為每一個守護進程提供獨立的 UID/GID 的習慣。

步驟五：其它必要調整
　　如果在 named.conf 中還指定了另外的目錄和文件，也要相應地在 chroot()環境中（在本例中即/chroot/bind/目錄）進行設置。

步驟六：調試

1、終止系統中原有的 syslogd 和 named 守護進程。

killall syslogd named

2、用適當的參數重新啟動 syslogd 守護進程。
syslogd -s -p /chroot/bind/var/run/log

3、用適當參數重新啟動 named 守護進程。
/chroot/bind/named -u bind -g bind -t /chroot/bind

4、檢查 syslogd/named 守護進程、監聽連接埠是否正常，和/var/log/messages 文件中 named 進程啟動時是否正常。

ps auwx|grep syslogd
root 5896 0.0 1.7 896 508 ?? Ss 9:44PM 0:00.10 syslogd -s -p
/chroot/bind/var/run/log
ps auwx|grep named
bind 5941 0.0 4.9 1652 1444 ?? Is 9:52PM 0:00.01
/chroot/bind/usr/sbin/named -u bind -g bind -t /chroot/bind
netstat -an|grep 53
tcp4 0 0 127.0.0.1.53 *.* LISTEN
tcp4 0 0 192.168.8.19.53 *.* LISTEN
udp4 0 0 127.0.0.1.53 *.*
udp4 0 0 192.168.8.19.53 *.*

步驟七：修改系統啟動腳本

對於 FreeBSD 系統，在/etc/rc.conf 文件中增加如下內容即可：

```
syslogd_enable="YES"
# 如果希望禁止向外送出日志，將-s 換成-ss。
syslogd_flags="-s -p /chroot/bind/var/run/log"
named_enable="YES"
named_program="/chroot/bind/usr/sbin/named"
named_flags="-u bind -g bind -t /chroot/bind"
```

注：如果在其它系統平台，如 OpenBSD、Linux、Solaris，則可能會稍有不同。主要是不同平台上的 syslog 實現不盡相同。例如對於 OpenBSD 和 Linux 系統，打開日志別名 socket 的命令是"syslogd -a /chroot/bind/var/run/log"，而 Solaris 的 syslogd 守護進程則不支持別名。因此 Solaris 系統平台上的 chroot 需要通過另外的方法實現，關於具體的實現過程我會在另外的文章中說明。

---[[結束語]]---

安全增強配置的文章寫完了，但並不是說只要你按本文提到的方法和技術實施就能萬無一失，高枕無憂了。其實以上設置對 NXT 和 TSIG 遠程漏洞攻擊並不沒太多的防御作用，充其量只不過是要編寫更多的 shellcode 代碼來突破 chroot 環境的限制。即使用 allow-query 等極其嚴格地限制查詢客戶端（實際上在互聯網上並不現實），基於 UDP 協議的 TSIG 攻擊技術也只需構造偽造 IP 地址的資料包即可繞過其限制。

所以，在對 BIND（還有其它應用服務）進行安全增強配置的基礎上，安全管理員仍然需要密切關注最新的安全公告、安全補丁和安全技術，經常與專業的電腦安全專家交流知識和經驗，再輔以必要的安全產品和安全服務，才能更充分地保護好自己的網絡和電腦使用者，抵御各種惡意攻擊。

附錄二 BIND 9.X 的安全設定

作者：wuming < mailto:wuming@geekbone.org >
主頁：http://geekbone.org
日期：2001-5-15

---[[前言]]---

由於作者工作中管理數個域名服務器，在最近工作當中對 BIND9 和 BIND8 進行了一些比較。從中的到了一些心得體會，在看過 backend 的“BIND 8+ 域名服務器安全增強”一文后，發覺國內的確還沒有太多關於 DNS 服務安全配置方面的較為完整的文章，在這里拙筆翻譯和寫一些我所了解如何安全配置 BIND，希望能給系統管理員一些幫助。

在此文章中將介紹如何在兩台 LINUX（內核：2.2.x 和 2.4.x）機器下安裝 chroot 過的 BIND 版本 9。同時還舉出一些如何設置 TSIG 的例子等等。通過這篇文章希望能夠幫助使用 BIND8 的管理員向 BIND9 移植。

---[[內容簡介]]---

---[[概況，BIND8 和 BIND9 的區別]]---

按照 ISC 的調查報告，BIND 是世界上使用最多最廣泛的域名服務系統。不論你的信件服務器，WEB 服務器或者其他的 services 如何的安全可靠，DNS 的故障會給你帶來使用者根本無法訪問這些服務。

BIND，也是我們常說的 named，由於多數網絡應用程序使用其功能，所以在很多 BIND 的弱點及時被發現。主要分為三個版本：
1：v4，1998 年多數 UNIX 捆綁的是 BIND4，已經被多數廠商拋棄了，除了 OpenBSD 還在使用。OpenBSD 核心人為 BIND8 過於復雜和不安全，所以繼續使用 BIND4。這樣一來 BIND8/9 的很多優點都不包括在 v4 中。

2：v8，就是如今使用最多最廣的版本，其詳細內容可以參閱“BIND 8+ 域名服務器安全增強”

3：v9，最新版本的 BIND，全部重新寫過，免費（但是由商業公司資助），也添加了許多新的功能（但是安全上也可能有更多的問題）。BIND9 在 2000 年十月份推出，現在穩定版本是 9.1.2。

幾家大公司對 BIND9 的評價十分高，稱 BIND9 十分專業和把 DNS community 提昇到一個新的高度。
我們是否應該注意 DNS 的安全問題？當然，一個管理不善的 DNS 可能帶來以下一些值得一提危

險。

1，如果任意允許 zone transfer，一個攻擊者能夠輕而易舉的得到有關 zone 的資訊，這樣其中比如 route，其他重要 hosts/intern hosts 都被泄露。

2，Denial of service，拒絕服務攻擊可能帶來危害：

·你的網站不能訪問，也不能訪問其他網站，因為沒有域名解譯。
·信件不能接收（雖然一些網站有快取，但是不會超過幾個小時或者一兩天時間。）
·攻擊者可以偽裝 DNS 服務提供假的 DNS 資訊。這會帶來什麼后果？

3，徹底沒有防護：如果攻擊者能夠偽造你的 DNS 資料或者欺騙其它網站相信假的 DNS 數據（稱 DNS poisoning），后果不堪設想……

+ 假造你的網頁，並且很容易的騙出使用者名和密碼，或者其他敏感資料。
+ 所有的信件到達你的服務器之前可以被拷貝，修改或者刪除。
+ 如果使用防火墻或者其他服務涉及 DNS 域名相信(auth)的服務，將徹底沒有任何防護作用。比如一個網絡代理只允許 *.mydomain.com 訪問。攻擊者很容易添加他自己的域名。對於防火墻比如只通過 admin.mydomain.com 進入，那麼入侵者添加其 IP 就可以了。

2000 年底由於多個 BIND 遠程溢出，使 DNS 成為黑客最喜好攻擊的目標之一。應該做些什麼？BIND 主要有哪些危險和弱點？

1，首先應該隔離 BIND 服務器，不應該在 DNS 服務器上跑其他服務，尤其是允許普通使用者登陸。減少其它的服務可以縮小被攻擊的可能性，比如混合攻擊。

2，雙 DNS 服務器，第二個(secondary DNS)應該安裝在另外一個網絡連接上，（不走同一個 ISP 等）。如果你的 DNS 服務器因為某種原因斷線，至少還有一個快取。這樣一來比如 EMAIL 等都不會丟失。一般可以維持四天(應該能夠修好了吧？！)。

3，使用最新版本的 BIND！（比如 8.2.3 或者 9.1.2）

4，權限管理（Access Control），限制 zone transfer 的範圍，不給攻擊者得到你局域網的資訊。可以使用傳遞簽名(transaction signatures)等。

5，用最低的權限執行 BIND，即非 root 使用者和嚴格的 umask 設置

6，使用 chroot 對執行 BIND 時添加更多的隔離，這樣 BIND 對系統和其他服務帶來修改和破坏會更困難。

7，雖然有些人不相信隱蔽 BIND 的序號對安全有什麼特別的好處，可以有效的組織大多數 script kiddie 的掃瞄。對於專業的駭客是另外一回事。

8，BIND 的紀錄應該常常分析，對於不尋常的記錄可以使用內設的 checker。

9，backend 說：對 BIND（還有其它應用服務）進行安全增強配置的基礎上，安全管理員仍然需要密切關注最新的安全公告、安全補丁和安全技術，經常與專業的電腦安全專家交流知識和經驗。

文章將具體介紹 4，5，6，應該能夠達到相對的安全高度。

BIND9 還是 BIND8？
一些值得一提的區別是：

·如果 named.conf 有語法錯誤，BIND9 會記錄錯誤和繼續執行域名服務，而 BIND8 只會記錄錯誤和 segmentation fault 或者 core dump！
·對於事務簽名(TSIG)比 BIND8 有更好的支持。
·新的 start/stop/reload 工具等等。比如 rndc，附有新的 Domain 更新方式。
·zone 文件語法檢測，比如 TTL 行必須存在。
·named.conf 中：
BIND8 中的 check-name 和 statistics-interval 不在使用
缺省屬性 auth-nxdomain 自動為"no"，而 BIND8 是"yes"
·不需要 root 服務器列表，就是 named.boot 或者 root.hint 包含在 server 中了。
·可以使用多進程域名服務（在你的機器 threads 支持良好條件下），如：
% ps ax |grep named
3947 ? S 0:00 /usr/local/sbin/named -u nobody
3948 ? S 0:00 _ /usr/local/sbin/named -u nobody
3949 ? S 0:00 _ /usr/local/sbin/named -u nobody
3950 ? S 0:00 _ /usr/local/sbin/named -u nobody
·BIND9 不支持名字認定，這樣一來你就有可能設置更多的域名，比如下划線。但還是不一定能解釋帶有怪字符域名。
·更多請參閱 http://sysadmin.oreilly.com/news/dnsandbind_0401.html

---[[安裝和設置 BIND]]---
以下是我如何在 LINUX 機器上安裝和設置 BIND9 的過程，首先檢查你是否有 gcc，ass 等 BIND 需要軟體。

1，編譯 BIND
下載最新的 BIND，解開壓縮，編譯。做這一步不需要 root 權限。(建議先安裝 GNU make，如果你沒有的話）首先在主域名服務機器上安裝。

./configure --disable-threads
（對於差一點的機器不建議使用 threads 功能）
一般來說：
./configure

然后就可以編譯了：
make

現在變成 root，暫時將 BIND 裝在臨時目錄(比如/tmp/)，並且生成一個 tarball(塔塔球？)。
```
su
umask 027
#生成文件權限
make install DESTDIR=/tmp

#注意，在這里我使用了 strip 指令，並不一定需要，只是將所有 Debug 資料刪
除，能將本身約 60MB 的 BIND 減肥成 18MB 左右。(版本 9.1.2)
cd /tmp/usr/local/
strip bin/* sbin/* lib/*
rm -rf include
#刪除 include 也是可選擇的，我是為了減少空間。
tar -cf - * | gzip > bind9.tar.gz
```

現在可以把 bind9.tar.gz 移到一個安全的目錄，並且刪除在/tmp 不必要的文件。
BIND 文獻可以在 doc/arm/Bv9ARM.html 找到，有時間值得一讀。

2，設置 chroot 和安裝 BIND

chroot() 設置是為了是 BIND 更加安全，使入侵者即使攻入也無法閱讀任何系統文件，比如多數允許匿名 FTP 使用 chroot()。

1，首先指定你在哪里安裝 BIND，指定權限應該是 022。
```
export jail='/usr/local/dns'
umask 022
```

2，建立目錄和連接 chroot 運行環境。
```
mkdir $jail
cd $jail
mkdir -p {dev,usr,var,etc};
mkdir -p var/{run,log,named} usr/lib;
mkdir -p usr/local/etc/
mkdir -p usr/share/lib/zoneinfo
```

3，建立帳號
```
groupadd named
useradd -d /tmp -s /bin/false -g named -c "BIND daemon" -m named
#在 chroot 目錄下生成必要 passwd 等文件。
grep named /etc/passwd >> $jail/etc/passwd
grep named /etc/shadow >> $jail/etc/shadow
grep named /etc/group >> $jail/etc/group

#禁止 BIND 帳號使用 FTP
echo "named" >>/etc/ftpusers
```

#也許你 ftpusers 在別的目錄下。

4，安裝編譯好的 BIND，首先拷貝到 chroot() 目錄下。
cp bind9.tar.gz $jail
cd $jail
tar -xvzf bind9.tar.gz

5，拷貝為 chroot 環境的系統文件
cp /etc/{syslog.conf,nsswitch.conf,resolv.conf,timezone} $jail/etc

6，查看 named 需要哪些 lib 文件
ldd /usr/local/dns/sbin/named
#將這些文件拷貝到 lib 目錄下，在 Linux 下例如：
cp -p /lib/libnsl.so.1 \
/lib/libpthread.so.0 /lib/libc.so.6 \
/lib/ld-linux.so.2 /usr/local/dns/lib
(如果拷貝了以上文件后 BIND 不工作，根據“經驗”以下的文件最好拷貝到 lib
木錄下。)
cp /lib/ld.so* /lib/libnss_files* /usr/local/dns/lib/

7，設置時區和其他系統設置。
mkdir -p /usr/local/dns/usr/share/zoneinfo;
cp -p /usr/share/zoneinfo/MET /usr/local/dns/usr/share/zoneinfo/MET
cd /usr/local/dns/dev
mknod tcp c 11 42
mknod udp c 11 41
mknod log c 21 5
mknod null c 13 2
mknod zero c 13 12
chgrp sys null zero
chmod 666 null
mknod consolg c 21 0
mknod syscon c 0 0
chmod 620 syscon
chgrp tty syscon
chgrp sys consolg
chgrp sys conslog

8，生成域名資料文件
mkdir -p /usr/local/dns/var/named;

3，設置 DNS 資料
cd /usr/local/dns/etc/
touch named.conf

chown root:named named.conf
chmod 640 named.conf

vi named.conf
#你完全可以使用其它編輯器我們應該在 named.conf 里面寫些什麼？在這里做一個簡單例子，比如如何設置 geekbone.org 這個域名。

1：主 DNS 服務器，假設地址為 192.168.1.3
在 /usr/local/dns/var/named 下生成 geekbone.zone, geekbone.rev,
dns.geekbone.org
以及 named.conf.primary（以上文件分別可以在這篇文章末尾找到，以免產生疑惑。）

2：副 DNS 服務器，假設地址為 192.168.1.4
生成以下幾個文件， geekbone.zone, geekbone.rev 和 named.conf.secondary

下面我會具體介紹一些如何設置 BIND，首先我們拷貝好和修改好這些文件后，用 BIND 工具檢查語法。
/usr/local/dns/sbin/named-checkconf /usr/local/dns/etc/named.conf
/usr/local/dns/sbin/named-checkzone
/usr/local/dns/var/named/geekbone.rev
/usr/local/dns/sbin/named-checkzone
/usr/local/dns/var/named/geekbone.zone
/usr/local/dns/sbin/named-checkconf
/usr/local/dns/var/named/dns.geekbone.org
#注意，很多中國網管沒有.rev 的權限，故可忽略 rev 和 zone 文件。

4，設置“監獄”模式
首先設置 named 必要的權限，並且禁止 SUID/SGID 文件等。以下是如何安全運行 BIND 最過分的一種做法，我們稱 PARANOID 模式。管理員應當根據服務器情況而設置。

cd /usr/local/dns/
chgrp -R named *
chmod -R g-w var;

#在兩個 DNS 服務器上，只需要允許建立和修改文件權限。
chown -R root:named /usr/local/dns/var/named/
chmod 770 /usr/local/dns/var/named/

#生成空 log 和 pid 文件：
touch /usr/local/dns/var/log/all.log
/usr/local/dns/var/run/named.pid
chown named:named /usr/local/dns/var/log/all.log
/usr/local/dns/var/run/named.pid

```
#給 named 權限寫 log 和 pid 文件
chgrp -R named /usr/local/dns/var/log/ /usr/local/dns/var/run/
chmod 770 /usr/local/dns/var/log/ /usr/local/dns/var/run/
chmod -R o-rwx /usr/local/dns/var/log/ /usr/local/dns/var/run/

#給 named 訪問其他設置文件
chgrp named /usr/local/dns/etc
chown root:named /usr/local/dns/etc/named.conf
chmod 640 /usr/local/dns/etc/named.conf
chmod 755 /usr/local/dns/etc

#檢查 BIND 目錄，查看是否有 SUID 和 SGID 程序，如果除去 SUID 或 SGID 權限
find /usr/local/dns/ -type f -exec chmod ug-s {} \;

#除去普通使用者閱讀權限
chmod -R o-rwx * /usr/local/dns/usr/
```

5，啟動 BIND

OKEY！我們已經將 BIND 設置好了，下面讓我們試著啟動 BIND！

```
1，以下兩條指令根據 syslog.conf 設置不同，種類不一。
tail -f /var/log/messages | grep named &
tail -f /var/log/daemon.log | grep named &
#為了方便觀察 BIND 啟動時所發生的事情。
2，啟動 named
/usr/local/dns/sbin/named -t /usr/local/dns/ -u named
#-t 表示 chroot 的目錄 -u 表示使用者

3, 檢查錯誤，如果你發現一些設置錯誤，修改並且重新啟動。
kill -1 `cat /usr/local/dns/var/run/named.pid`

4，如果一切正常，那麼恭喜你的 BIND 安裝成功了！下一步就是在系統啟動
的時候自動啟動 BIND，這里有設置的一個 BIND 啟動器。/etc/init.d/dns，
一般只需要連接到 /etc/rc2.d/S50dns 和 /etc/rc2.d/K50dns 就可以了。
```