输入关键字

DNS原理介绍及Centos/Redhat/Fedora系统下用bind和bind-chroot的正反向解析域名的配置 (https://www.cnhzz.com/bind-dns-base/)

2014-05-06 分类：服务应用 (https://www.cnhzz.com/category/server/) 阅读(1283) 评论(0)

在访问一个网站的时候，只要输入该网站的网址就会跳转到该网站页面，而实现这一过程就需要DNS服务器将域名解析为网站服务器IP地址，进而实现数据通信。那么DNS服务器是如何工作的呢？本文分为两部分，本文将详解DNS服务原理及正反向解析配置。

DNS服务原理介绍

域名解析服务：DNS（Domain Name Service）

应用程序：bind

监听端口：udp/53，tcp/53（一般TCP/53用于同步主从DNS服务器的数据信息， 而UDP/53则用于提供Client的DNS查询服务）

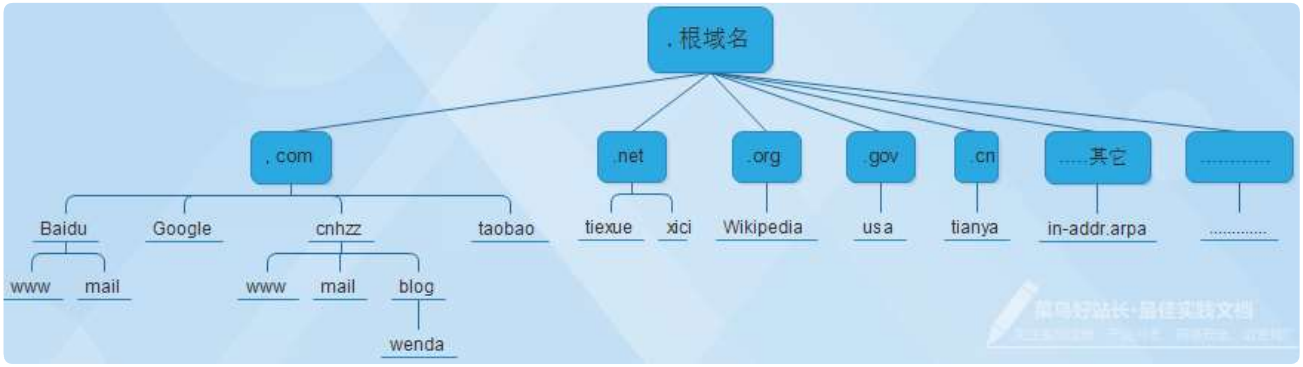
根域：. (半角的点.)

根服务器：从A至M编号，其中：美国10个(1个主根和9个辅根)、欧洲2个(位于英国和瑞典)、亚洲1个(位于日本)。1个为主根服务器，放置在美国弗吉尼亚州的杜勒斯，由美国VeriSign公司负责运营维护，其余12个均为辅根服务器。

一级域(顶级域)：

- 组织域：.com, .org, .net, .mil, .edu, .gov, .info, .cc, .me, .tv
- 国家域：.cn, .us, .uk, .jp, .tw, .hk, .iq, .ir
- 反向域：.in-addr.arpa

DNS树状结构:



解析方式:

FQDN:(Fully Qualified Domain Name)全称域名

例如：cnhzz.com

正向解析：FQDN –> IP

反向解析：IP –> FQDN

DNS查询方式

递归：一般客户机和服务器之间属递归查询，即当客户机向DNS服务器发出请求后,若DNS服务器本身不能解析,则会向另外的DNS服务器发出查询请求，得到结果后转交给客户机，（DNS请求被服务器接受后，如果属于此服务器管辖范围则请求上级服务器依次传递请求，并且依次传递结果给发出请求的主机。）

迭代： DNS请求被服务器接受后，如果不是自己管辖范围，让客户端访问根域服务器，然后跟域通知客户端去访问下级服务器，直到最后客户端访问管辖请求域名的服务器为止。（一般DNS服务器之间属迭代查询）

因为迭代查询极耗资源，所以DNS服务器通常只为自己组织内的客户端做解析，通过在权限上加以限制来实现

区域解析库:

资源记录：rr(resource record)用于此记录解析的属性

SOA: Start Of Authority, 起始授权记录，一个区域文件只能有一个

NS：Name Server，域名服务器

MX: Mail eXchange, 邮件交换器，MX记录有优先级属性（0-99）

A：FQDN –> IP，专用于正向解析库 （FQDN：Full Qualified Domain Name–完全限定域名）

PTR: IP –> FQDN，专用于反向解析库（ 反向解析难度极大，目前应用：邮件服务器根据反向解析，拒绝接收来自没有域名的站点发来的信息以降低垃圾邮件的数量）

AAAA：FQDN –> IPv6，专用于正向解析库

CNAME: Canonical Name，正式名称

> 注意：正、反向解析是两个不同的名称空间，是两棵不同的解析树；各需要一个解析库来分别负责本地域名的正向和反向解析

DNS服务器类型:

主DNS服务器

从（辅助）DNS服务器

缓存DNS服务器

转发器

主DNS服务器：维护所负责解析的域内的解析库（解析库由管理员维护）

- 序列号：解析库的版本号；同步解析库的前提：主服务器解析库内容发生变化，其序列号递增加1
- 刷新时长：从服务器向主服务器请求同步解析库的时间间隔
- 重试时长：从服务器向主服务器请求同步解析库失败时，再次尝试的时间间隔
- 过期时长：从服务器始终联系不到主服务器时，多久之后放弃从服务器角色，停止服务
- 否定应答的TTL：解析得到的否定答案的缓存时长

从DNS服务器：解析库从主DNS服务器或其他DNS服务器“复制”（区域传送）而来

缓存DNS服务器：不负责解析任何域，也不用注册域名，用于帮助域内的客户端向互联网的DNS服务器发送请求，把结果返回给客户端的同时在服务器上做一份缓存

区域传送:

解析库文件同步的过程，即辅助DNS服务器与主DNS服务器间的区域文件同步传输过程。

完全区域传送：传送区域的所有数据，AXFER

增量区域传送：传送区域中改变的数据，IXFER

DNS资源记录格式

```
格式：
name    [ttl]  IN      RRType     value
```

SOA记录：

```
name: 区域名称，通常可以简写为@
value: 主DNS服务器的FQDN，也可以当前区域的区域名称
注意：任何解析库文件的第一个记录的类型必须是SOA

例如：
@ IN SOA ns.cnhzz.com. root.cnhzz.com. (
 serial number ;#解析库版本号，例如2015040701
 refresh time ;#刷新时间，即同步时间
 retry time ;#重试时间
 expire time ;#过期时间
 negative answer ttl ;#否定答案的统一缓存时长
 )
```

NS记录：

```
name: 区域名称
value: DNS服务器的FQDN
注意：如果有多台NS服务器，每一个都必须有对应的NS记录；
对于正向解析文件来讲，每一个NS的FQDN都应该有一个A记录；

例如：
@ IN NS ns.scholar.com.
```

MX记录：

```
name: 区域名称
value: 邮件服务器的FQDN
优先级：0-99.数字越小，越优先
注意：如果有多台MX服务器，每一个都必须有对应的MX记录；但各MX记录还有优先级属性
对于正向解析文件来讲，每一个NS的FQDN都应该有一个A记录；

例如：
@ IN MX 10 mail.scholar.com.
@ IN MX 20 mail2.scholar.com.
```

A记录：

```
name: FQDN（可简写）
value: IP

例如：
www IN A 192.168.35.141
www IN A 192.168.35.141
pop3 IN A 192.168.35.141
imap IN A 192.168.35.141
```

AAAA记录：

```
name: FQDN（可简写）
value: ipv6 IP
```

CNAME记录：

```
name: FQDN
value: FQDN

例如：
www IN A 172.16.10.10
ftp IN CNAME www
```

PTR记录：

```
name: 逆向的主机IP地址加后缀in-addr.arpa，例如192.168.35.141，
网络地址为192.168，
主机地址为35.141，
其name为141.35.in-addr.arpa.（可简写）
value: FQDN

例如：
35 IN PTR www.cnhzz.com.
```

DNS服务配置和正反向解析

DNS服务程序包介绍

Centos/Redhat/Fedora 下可提供DNS服务的程序包为bind（bekerley internet name domain）

作为DNS服务器建议安装：`yum install bind*`

bind安装好之后主要的daemon是named，一般情况下会自动安装好bind-chroot，chroot的存在主要就是为了保护系统的安全性，就算bind被黑了，黑客也只能在chroot的目录里面活动。

启动：`service named restart`

```
================================================================================
 软件包                  架构          版本                          仓库          大小
================================================================================
正在安装:
 bind                    x86_64        32:9.8.2-0.30.rc1.el6_6.2     updates      4.1 M
 bind-chroot             x86_64        32:9.8.2-0.30.rc1.el6_6.2     updates       73 k
 bind-devel              x86_64        32:9.8.2-0.30.rc1.el6_6.2     updates      381 k
 bind-dyndb-ldap         x86_64        2.3-6.el6_6                   updates       71 k
 bind-sdb                x86_64        32:9.8.2-0.30.rc1.el6_6.2     updates      310 k
 bind-to-tinydns         x86_64        0.4.3-15.20140818gitdf0ddc3.el6
                                                                     epel          18 k
为依赖而安装:
 portreserve             x86_64        0.0.4-9.el6                   base          23 k
 postgresql-libs         x86_64        8.4.20-
2.el6_6                                updates      202 k

事务概要
================================================================================
Install       8 Package(s)

总下载量：5.1 M
Installed size: 11 M
确定吗？[y/N]：
```

服务脚本： `/etc/init.d/named`

不带有 chroot的主配置文件： `/etc/named.conf` ， `/etc/named.rfc1912.zones`

不带有 chroot的区域解析库文件： `/var/named/zone_name.zone`

有 chroot的主配置文件： `/var/named/chroot/etc/named.conf` ， `/var/named/chroot/etc/named.rfc1912.zones`

有 chroot的区域解析库文件： `/var/named/chroot/var/named/zone_name.zone`

主配置文件中通常有三个区域：根、localhost、127.0.0.1的反向区域

域类型：主域（master）、从域（slave）、缓存域（hint）、转发域（forward）

bind是linux的DNS服务器程序。bind–chroot是bind的一个功能。使bind可以在一个chroot 的模式下运行.也就是说,bind运行时的/(根)目录,并不是系统真正的/(根)目录,只是系统中的一个子目录而已.这样做的目的是为了提高安全性.因为在chroot的模式下,bind可以访问的范围仅限于这个子目录的范围里,无法进一步提升,进入到系统的其他目录中.

chroot监牢技术：chroot可以改变程序运行时所参考的根目录(/)位置，即将某个特定的子目录作为程序的虚拟根目录，并且对程序运行时可以使用的系统资源，用户权限和所在目录进行严格控制，程序只在这个虚拟的根目录下具有权限，一旦跳出该目录就无任何权限。

例如在RHEL5中：

/var/name/chroot实际上是根目录(/)的虚拟目录，

所以虚拟目录中的/etc目录实际上是/var/named/chroot/etc目录，

而/var/named目录实际上是/var/named/chroot/var/named目录。

chroot功能的优点是：如果有黑客通过Bind侵入系统，也只能被限定在chroot目录及其子目录中，其破坏力也仅局限在该虚拟目录中，不会威胁到整个服务器的安全。

实践Bind正反向解析域名配置

解析要求：要求可以实现正反向解析

服务器	IP地址	应用服务	完全限定域名（FQDN）
Server A	192.168.35.155	Mail	mail.cnhzz.com
Server B	192.168.35.134	Web, FTP	www.cnhzz.com，ftp.cnhzz.com
Server C	192.168.35.143	Primary DNS server	ns1.cnhzz.com

修改主DNS服务器的主机名

```
#修改主机名
vim /etc/sysconfig/network
HOSTNAME=ns1.cnhzz.com
#立即生效
hostname ns1.cnhzz.com
```

编辑主配置文件，添加正向区域和反向区域

拷贝bind相关文件,准备bind chroot 环境

```
cp -R /usr/share/doc/bind-9.8.2/sample/var/named/* /var/named/chroot/var/named/
```

将 /etc/named.conf /etc/named.rfc1912.zones 拷贝到 bind chroot目录（在安装bind-chroot的情况下，配置文件保存在/var/named/chroot/etc/目录下。）

```
cp -p /etc/named.conf /var/named/chroot/etc/named.conf
```

```
cp -p /etc/named.rfc1912.zones /var/named/chroot/etc/named.rfc1912.zones
```

编辑配置文件

```
vim /var/named/chroot/etc/named.conf
```

```
options {
    //listen-on port 53 { 127.0.0.1; }; //改成any或者注释掉
    //listen-on-v6 port 53 { ::1; }; //改成any或者注释掉
    directory   "/var/named"; //区域文件目录
    dump-file   "/var/named/data/cache_dump.db"; //#默认服务器存放数据库文件
        statistics-file "/var/named/data/named_stats.txt"; //#默认统计信息路径
        memstatistics-file "/var/named/data/named_mem_stats.txt"; //#默认内存使用统计文件
    //allow-query     { localhost; };//可访问控制主机，改成any或者注释后允许所有IP访问
        //allow-query-cache { any; }; //DNS缓存
    recursion yes; //#是否允许递归查询

    dnssec-enable yes;
    dnssec-validation yes;
    dnssec-lookaside auto;

    /* Path to ISC DLV key */
    bindkeys-file "/etc/named.iscdlv.key";

    managed-keys-directory "/var/named/dynamic";
};

logging {
        channel default_debug {
                file "data/named.run";
                severity dynamic;
        };
};

zone "." IN {
    type hint;
    file "named.ca";
};

include "/etc/named.rfc1912.zones";
include "/etc/named.root.key";
```

在bind chroot 的目录中创建相关文件

```
#创建相关文件
touch /var/named/chroot/var/named/data/cache_dump.db
touch /var/named/chroot/var/named/data/named_stats.txt
touch /var/named/chroot/var/named/data/named_mem_stats.txt
touch /var/named/chroot/var/named/data/named.run
mkdir /var/named/chroot/var/named/dynamic
touch /var/named/chroot/var/named/dynamic/managed-keys.bind
#将 Bind 锁定文件设置为可写
chmod -R 777 /var/named/chroot/var/named/data
chmod -R 777 /var/named/chroot/var/named/dynamic
```

到此基本上DNS服务器是可以正常运行了，别忘记去需要使用自建DNS服务器的应用服务器上把 `vim /etc/resolv.conf` 修改为自己这台服务器的IP！

开始创建域以及域文件（添加正向区域和反向区域）

可以直接修改named.conf，也可以在 named.rfc1912.zones下面创建，按照标准，建议在named.rfc1912.zones下面创建区域文件。

```
vim /var/named/chroot/etc/named.rfc1912.zones
```

```
zone "cnhzz.com" IN {                                   #正向区域
        type master;
        file "cnhzz.com.zone";
};

zone "35.168.192.in-addr.arpa" IN {                     #反向区域
        type master;
        file "192.168.35.zone";
};
```

正向区域解析文件

```
vim /var/named/chroot/var/named/cnhzz.com.zone
```

```
$TTL 3600
@       IN      SOA     ns1.cnhzz.com. root.cnhzz.com. (
                                                        2015040701      ;版本
                                                        2H              ;刷新
                                                        10M             ;重试
                                                        7D              ;过期
                                                        1D )            ;最小时间
@       IN      NS      ns1
@       IN      MX 10   mail
ns1     IN      A       192.168.35.143
mail    IN      A       192.168.35.155
www     IN      A       192.168.35.134
pop     IN      CNAME   mail
ftp     IN      CNAME   www
```

反向区域解析文件

```
vim /var/named/chroot/var/named/192.168.35.zone
```

```
$TTL 3600
@       IN      SOA     ns1.cnhzz.com.        root.cnhzz.com. (
                                                        2015042917
                                                        2H
                                                        10M
                                                        7D
                                                        1D )
@       IN      NS      ns1.cnhzz.com.
143     IN      PTR     ns1.cnhzz.com.
155     IN      PTR     mail.cnhzz.com.
134     IN      PTR     www.cnhzz.com.
```

检查配置，启动服务并且设置自启

```
[root@dns1 named]# service named configtest
zone localhost.localdomain/IN: loaded serial 0
zone localhost/IN: loaded serial 0
zone 1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.ip6.arpa/IN: load
ed serial 0
zone 1.0.0.127.in-addr.arpa/IN: loaded serial 0
zone 0.in-addr.arpa/IN: loaded serial 0
zone cnhzz.com/IN: loaded serial 2015040701
zone 35.168.192.in-addr.arpa/IN: loaded serial 2015042917
[root@dns1 named]# service named start
启动 named:                                                [确定]
[root@dns1 named]# ss -tunl | grep :53
udp    UNCONN     0      0         192.168.35.143:53                    *:*
udp    UNCONN     0      0              127.0.0.1:53                    *:*
tcp    LISTEN     0      3         192.168.35.143:53                    *:*
tcp    LISTEN     0      3              127.0.0.1:53                    *:*
   [root@dns1 named]# chkconfig named on
```

正向解析测试

```
dig -t A www.cnhzz.com @192.168.35.143
```

```
[root@dns1 ~]# dig -t A www.cnhzz.com @192.168.35.143

; <<>> DiG 9.8.2rc1-RedHat-9.8.2-0.30.rc1.el6_6.2 <<>> -t A www.cnhzz.com @192.168.35.
143
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 37742
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

;; QUESTION SECTION:
;www.cnhzz.com.         IN  A

;; ANSWER SECTION:
www.cnhzz.com.      3600    IN  A   192.168.35.134

;; AUTHORITY SECTION:
cnhzz.com.      3600    IN  NS  dns.cnhzz.com.

;; ADDITIONAL SECTION:
dns.cnhzz.com.      3600    IN  A   192.168.35.134

;; Query time: 1 msec
;; SERVER: 192.168.35.143#53(192.168.35.143)
;; WHEN: Thu Apr 30 17:37:36 2015
;; MSG SIZE  rcvd: 81
```

反向解析测试

```
dig -x 192.168.35.134 @192.168.35.143
```

```
[root@dns1 named]# dig -x 192.168.35.134 @192.168.35.143

; <<>> DiG 9.8.2rc1-RedHat-9.8.2-0.30.rc1.el6_6.2 <<>> -x 192.168.35.134 @192.168.35.1
43
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 42693
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 1, ADDITIONAL: 1

;; QUESTION SECTION:
;134.35.168.192.in-addr.arpa.   IN  PTR

;; ANSWER SECTION:
134.35.168.192.in-addr.arpa. 3600 IN    PTR dns.cnhzz.com.
134.35.168.192.in-addr.arpa. 3600 IN    PTR www.cnhzz.com.

;; AUTHORITY SECTION:
35.168.192.in-addr.arpa. 3600   IN  NS  dns.cnhzz.com.

;; ADDITIONAL SECTION:
dns.cnhzz.com.      3600    IN  A   192.168.35.134

;; Query time: 0 msec
;; SERVER: 192.168.35.143#53(192.168.35.143)
;; WHEN: Wed May  6 14:05:51 2015
;; MSG SIZE  rcvd: 120
```

经过dig命令的测试，说明DNS服务器已经搭建完毕了。

分享到：更多 ()

标签： DNS (https://www.cnhzz.com/tag/dns/)

相关技术点推荐

- bind配置 DNS子域授权和转发 (https://www.cnhzz.com/bind-glue-record/)
- Centos/Redhat/Fedora系统环境下的DNS主从服务器配置 (https://www.cnhzz.com/dns-master-slave/)
- HTTP Header Request和Response两部分详解 (https://www.cnhzz.com/http-header-request-response/)
- May 28 03:32:59 aliyun kernel: Killed process 8057 (sphinx-intl) total-vm:309212kB, anon-rss:103816kB, file-rss:0kB (https://www.cnhzz.com/may-28-033259-aliyun-kernel-killed-process-8057-sphinx-intl-total-vm309212kb-anon-rss103816kb-file-rss0kb/)
- 通信协议—IP、Http、TCP、UDP、Socket五者的关系: (https://www.cnhzz.com/ip-http-tcp-udp-socket/)
- Nginx查看并发链接数 (https://www.cnhzz.com/nginx-status/)
- 让你的 Nginx启用 HTTP/2 (https://www.cnhzz.com/nginx-http2/)
- Rewrite规则 在Nginx中的使用 (https://www.cnhzz.com/creating-nginx-rewrite-rules/)