DNS服务器

单一文件处理上网的年代hosts、分布式，阶层式主机名管理架构：DNS，完整主机名FQDN（Fully Qualified Domain Name）
[root@personal ~]# vim /etc/nsswitch.conf 决定先使用hosts还是resolv.conf

1、介绍
domain name system 或者 domain name service 。将域名转换为IP地址的功能的服务器，DNS使用的是tcp和udp 端口是53。主要使用的还是udp。tcp协议一般实现服务器之间的备份。当hosts和resolv.conf文件共存时，根据 /etc/nsswitch.conf 来决定哪种方法优先
2、DNS的域名结构
www.baidu.com.cn.　　主机名.域名(三级域名).域名(二级域名).域名(顶级域名).　最后的一个点叫根域
全世界最大的三个网络信息中心，分别是美国的inter-nic、荷兰的ripe-nic、日本的apnic。全球有13台根域服务器，从A到M排列。

3、dns查询
DNS的查询方式有两种：递归查询和迭代查询
递归查询：从根域到二级域再到三级域，服务器层级的进行查询获得的
- 当需要进行DNS解析时，系统向本地DNS服务器发出DNS解析请求，由本地的DNS服务器进行域名和IP地址的解析工作
- 本地DNS服务器收到用户的请求后，在自身的DNS数据库中进行查找匹配的域名和IP地址对应的记录。如果找到，把结果返回给客户端，并结束本次解析工作。如果没有找到，把请求发给根域DNS服务器
- 根域DNS服务器查找域名所对应的顶级域，再由顶级域找到二级域，由二级域查到三级域，以此类推。并把结果返回给本地DNS服务器
- 最终由本地DNS服务器把结果返回给客户端
- 如果经过查找后依然无法找到需要解析的记录，则由本地DNS服务器向客户端返回无法解析的错误信息

迭代查询：DNS服务器之间进行的

名称	管理单位	地理位置	IP地址
A	INTERNIC.NET	美国弗吉尼亚州	198.41.0.4
B	美国信息科学研究所	美国加利福尼亚州	128.9.0.107
C	PSINet公司	美国弗吉尼亚州	192.33.4.12
D	马里兰大学	美国马里兰州	128.8.10.90
E	美国航空航天管理局	美国加利福尼亚州	192.203.230.10
F	因特网软件联盟	美国加利福尼亚州	192.5.5.241
G	美国国防部网络信息中心	美国弗吉尼亚州	192.112.36.4
H	美国陆军研究所	美国马里兰州	128.63.2.53
I	Autonomica公司	瑞典斯德哥尔摩	192.36.148.17
J	VeriSign公司	美国弗吉尼亚州	192.58.128.30
K	RIPE NCC	英国伦敦	193.0.14.129
L	IANA	美国弗吉尼亚州	199.7.83.42
M	WIDE Project	日本东京	202.12.27.33

4、DNS服务器的安装
[root@test ~]# **yum install bind bind-chroot -y** 安装bind和bind-chroot
```
[root@test ~]# systemctl start named
[root@test ~]# systemctl enable named
```

5、chroot
出于安全方面的考虑，我们在安装dns服务器时同步安装了chroot的功能，对于named这个进程来说，它的根目录不在是默认的/，而变成了/var/named/chroot，这种做法可以提高系统的安全性，对于渗透者来说隐藏了真实的目录。（是否启动chroot及额外的参数：/etc/sysconfig/named）

```
[root@test ~]# vim /etc/named.conf
options {
    listen-on port 53 {any; };
};
```

配置正向、反向解析(基础)：
[root@RHEL6 /]# **vim /etc/named.conf** 修改文件，添加正向、反向解析的文件信息
```
zone "." IN {
    type hint;
    file "named.ca"; # 文件存在在/var/named/
```

```
};
zone "test.com" IN {
        type master;
        file "named.com";
};
zone "88.168.192.in-addr.arpa" IN {
        type master;
        file "named.com.arpa";
};
```

#、正向解析

[root@test ~]# **cp named.localhost named.com** //根据模板生成一个配置文件 ， 这个区域配置文件的名称必须跟注配置文件中的区域名称一致上图中的写法是完整写法，通常情况下不需要这样全部写明，我们可以简化它的写法

注：服务器的类型除了A记录外，还有CNAME（别名），PTR（反向记录），MX（电子邮件服务器），NS（DNS服务器）


```
[root@test ~]# cat /var/named/named.com
$TTL 1D
@       IN SOA @ root (
                                        0       ; serial
                                        1D      ; refresh
                                        1H      ; retry
                                        1W      ; expire
                                        3H )    ; minimum
        NS      @
        A       192.168.88.174
www     A       192.168.88.174
ftp     A       192.168.88.174
wpa     CNAME   www
```

[root@test named]# **chgrp named named.com** 必须修改区域文件的所属组，让named进程可以调用

[root@test named]# **vim /etc/resolv.conf** # 修改服务器的DNS地址

#、反向解析

[root@RHEL6 named]# **cp named.com named.com.arpa** 复制并生成一个反向解析的区域文件

[root@RHEL6 named]# **chgrp named named.com.arpa**

[root@RHEL6 named]# **vim named.com.arpa**


```
[root@test ~]# cat /var/named/named.com.arpa
$TTL 1D
@       IN SOA  test.com. root (
                                        0       ; serial
                                        1D      ; refresh
                                        1H      ; retry
                                        1W      ; expire
                                        3H )    ; minimum
        IN NS   test.com.
174     IN PTR  ns.test.com.
174     IN PTR  mail.test.com.
174     IN PTR  www.test.com.
```

客户端验证：[root@test named]# **dig -t a www.baidu.com.cn.**

客户端验证：[root@test named]# **dig -t ns www.baidu.com.cn.**
客户端验证：[root@test named]# **dig -x 192.168.24.111**
[root@test named]# nslookup

- RHEL7下的DNS服务器配置

RHEL7开始默认的DNS由unbound替代了bind。Unbound是一个非常轻量的DNS，改进了bind对DNS的一些特点，增加了防止DNS反射攻击的一些性能。RHEL7中也支持继续bind（不推荐），还支持使用dnsmasq（openstack中使用的DNS，更为强悍的DNS工具）。
[root@test /]# yum install unbound* -y　安装unbound
[root@test /]# systemctl restart unbound
[root@test /]# systemctl enable unbound
[root@test /]# netstat -ntulp |grep unbound
[root@test /]# firewall-cmd --add-service=dns
[root@test /]# firewall-cmd --add-service=dns --permanent
[root@test /]# vim /etc/unbound/unbound.conf

```
38          interface: 0.0.0.0          默认只监听自己的环回地址，去掉注释，监听所有网段
39          # interface: ::0
40          # interface: 192.0.2.153
```

```
176         access-control: 0.0.0.0/0 allow     放行所有网段的请求
```

```
211         username: " "
212                   默认提供认证机制，客户端请求需要提供用户名，取消用户名
```

[root@test local.d]# vim /etc/unbound/local.d/test.com.conf　生成一个区域解析文件，这个文件里面的内容也可以放在unbound.conf中，那样就可以注释掉inclue内容。

```
[root@test local.d]# cat test.com.conf
local-zone: "test.com." static
local-data: "test.com. 86400 IN SOA test.test.com. root.test.com. 100 1
00 200 400 86400 "
local-data: "test.test.com. 86400 IN A 192.168.24.11"
local-data: "www.test.com. 86400 IN A 192.168.24.11"
local-data: "ftp.test.com. 86400 IN A 192.168.24.11"
```

RHEL7下配置仅缓存DNS服务器
配置仅缓存服务器，只需要在unbound.conf中完成最基本的配置，不需要指明反向或正向区域，但需要在/etc/unbound/local.d/目录下创建一个conf文件
[root@test local.d]# vim cacha.conf

```
[root@test local.d]# cat cacha.conf
forward-zone:
        name: "test.com"
        forward-addr: 192.168.24.12
```

当仅缓存服务器配置好后，此时是无法获得正确的DNS解析地址的，原因是unbound比bind更注重安全性，默认进行转发时，两台DNS服务器在交换数据的过程中需要加密验证，这样的目的是为了防止DNS污染。所以如果两台DNS之间要正常通信，要么验证信息配置一致，要么关闭。以下操作必须在两台认证的服务器上都配置

```
367         #Ignore chain of trust. Domain is treated as insecure.
368         domain-insecure: "test.com"
```

如果需要验证：

```
354         trust-anchor: "nlnetlabs.nl. DNSKEY 257 3 5 AQPzzTWMz8qSWIQlfRnPckx2BiVmkVN6LP
    up03mbz7FhLSnm26n6iG9N Lby97Ji453aWZY3M5/xJBSOS2vWtco2t8CO+xe01bc/d6ZTy32DHchpW 6rDH1v
    p86Ll+haOtmwyy9QP7y2bVw5zSbFCrefk8qCUBgfHm9bHzMG1U BYtEIQ=="
355         # trust-anchor: "jelte.nlnetlabs.nl. DS 42860 5 1 14D739EB566D2B1A5E216A0BA4D1
    7FA9B038BE4A"
                                    启用这里的内容，并保证另一台服务器一致，这样两台服务器之间交流数据就会验证
                                    可以防止DNS污染
356
357         # File with trusted keys for validation. Specify more than one file
358         # with several entries, one file per entry. Like trust-anchor-file
359         # but has a different file format. Format is BIND-9 style format,
360         # the trusted-keys { name flag proto algo "key"; }; clauses are read.
361         # trusted-keys-file: ""
362         #                   注释掉下面的内容
363         # trusted-keys-file: /etc/unbound/rootkey.bind
364         #trusted-keys-file: /etc/unbound/keys.d/*.key
365         #auto-trust-anchor-file: "/var/lib/unbound/root.key"
366
367         #Ignore chain of trust. Domain is treated as insecure.
368         #domain-insecure: "test.com"
369
```

[root@test unbound]# dig +trace www.baidu.com 跟踪百度的DNS解析过程
[root@test unbound]# whois baidu.com

7、生成区域数据信息
[root@hf etc]# cd /var/named/chroot/var/named/
[root@hf named]# cp named.localhost test.com.ca //根据模板生成一个配置文件，这个区域配置文件的名称必须跟注配置文件中的区域名称一致

```
$TTL 1D
@       IN SOA  @ rname.invalid. (
                                        0       ; 序列号，定义主从服务器的更新
                                        1D      ; 刷新时间，默认是1天，从每隔一天向主同步
                                        1H      ; 重试时间，如果主不响应，从每隔1小时重试
                                        1W      ; 如果主不响应达到一周，那么不再重试
                                        3H )    ; 最小的缓存时间是3小时
        NS      @
        A       127.0.0.1
        AAAA    ::1

$TTL 1D     （存活期是1天）
test.com.       IN      SOA     hf.test.com.            root@test.com.          (
        （级别）（必须写）（这里的@也是一个变量，代表主机名，最后的点一定不能少）  （定义管理员邮箱）  0
        NS(name server,这个定义是DNS服务器类型，  hf.test.com.（指明域名服务器的名称）
            NS表明是域名服务器）
hf.test.com.  A  （A表示A记录，也是定义的类型，这个类型就是address）  192.168.24.244
www.test.com. A  192.168.24.234
wap.test.com. A  192.168.24.232
ftp.test.com. A 192.168.24.233
```

上图中的写法是完整写法，通常情况下不需要这样全部写明，我们可以简化它的写法
注：服务器的类型除了A记录外，还有CNAME（别名），PTR（反向记录），MX（电子邮件服务器），NS（DNS服务器）

```
$TTL 1D
@       IN SOA  hf.test.com.    root (
                                        0       ; 序列号，定义主从服务器的更新
                                        1D      ; 刷新时间，默认是1天，从每隔一天向主同步
                                        1H      ; 重试时间，如果主不响应，从每隔1小时重试
                                        1W      ; 如果主不响应达到一周，那么不再重试
                                        3H )    ; 最小的缓存时间是3小时
        NS      hf.test.com.
hf      A       192.168.24.244
www     A       192.168.24.244
ftp     A       192.168.24.220
wpa     CNAME   www
```

上图是简化后的写法，也就是标准的写法
[root@hf named]# chgrp named test.com.ca 必须修改区域文件的所属组，让namde进程可以调用

配置反向解析：
[root@hf /]# cd /var/named/chroot/etc/
[root@hf etc]# vim named.conf 修改文件，添加反向解析的文件信息

```
zone "." IN {
        type hint;
        file "named.ca";
};
zone "test.com" IN {
        type master;
        file "test.com.ca";
};
zone "24.168.192.in-addr.arpa" IN {
        type master;
        file "test.com.arpa";
};
```

[root@hf named]# cp test.com.ca test.com.arpa 复制并生成一个反向解析的区域文件
[root@hf named]# chgrp named test.com.arpa
[root@hf named]# vim test.com.arpa

```
1  $TTL 1D
2  @       IN SOA  hf.test.com.   root (
3                                       0       ; 序列号，定义主从服务器的更新
4                                       1D      ; 刷新时间，默认是1天，从每隔一天向
   同步
5                                       1H      ; 重试时间，如果主不响应，从每隔1小
   重试
6                                       1W      ; 如果主不响应达到一周，那么不再重试
7                                       3H )    ; 最小的缓存时间是3小时
8          NS      hf.test.com.
9  244     PTR     hf.test.com.
10 244     PTR     www.test.com.
11 220     PTR     ftp.test.com.
12
13
```

客户端设置dns后验证
服务器端验证
[root@hf /]# tcpdump -i eth0 port 53

- 主从DNS服务器配置

  为了保证服务高可用性，DNS要求可以使用多台服务器实现冗余。我们把为了实现冗余的服务器称为从DNS服务器，作用于同一个区域中主DNS服务器的备份。从DNS服务器定期与主DNS服务器进行区域信息的更新，保持一致。从DNS服务器会从主DNS服务器上获取最新区域的数据文件的副本，进行区域复制。

- 部署从DNS服务器

[root@nagios /]# yum install bind* -y
[root@nagios etc]# vim named.conf

```
37 zone "." IN {
38         type hint;
39         file "named.ca";
40 };
41
42 zone "test.com" IN {
43         type slave;
44         file "slaves/test.com.ca";
45         masters { 192.168.24.244; };
46 };
47
48 zone "24.168.192.in-addr.arpa" IN {
49         type slave;
50         file "slaves/test.com.arpa";
51         masters { 192.168.24.244; };
52 };
```

配置好从服务器后，我们需要在主服务器上添加条目，指定我们授权从DNS服务器，防止网络中的其他DNS从主DNS获取信息。
[root@hf etc]# vim named.conf 修改主服务器的DNS配置

```
options {
        listen-on port 53 { any; };
//      listen-on-v6 port 53 { ::1; };
        directory       "/var/named";
        dump-file       "/var/named/data/cache_dump.db";
        statistics-file "/var/named/data/named_stats.txt";
        memstatistics-file "/var/named/data/named_mem_stats.txt";
        allow-query     { any; };
        allow-transfer  { 192.168.24.220; };
        recursion yes;
```

- DNS安全之Transaction Signatures（事务数字签名TSIG）

  假设在网络上有一台恶意伪造从DNS的IP的服务器，它也希望通过主DNS服务器获取区域的副本。为了确保DNS之间消息的安全，我们可以在主从服务器之间配置TSIG（transaction signature）事务数字签名。它是通过共享密钥和单向散列函数的方式来验证DNS信息的。

  [root@test ~]# dnssec-keygen -a HMAC-MD5 -b 128 -n HOST test

Khf.+157+38164

```
[root@test ~]# ls
anaconda-ks.cfg       Ktest.+157+58478.key
initial-setup-ks.cfg  Ktest.+157+58478.private
```

使用dnssec-keygen命令生成了一个基于主域名服务器的共享密钥，一个是公钥，一个是私钥。我们指明的算法是HMAC-MD5的

[root@hf ~]# cat Khf.+157+38164.private 记录密钥的内容

将密钥的信息加入到主DNS服务器中

```
        allow-transfer  {  key hf; };
        recursion yes;

        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";
};

Server 192.168.24.220  {
         key  { hf; };
};
key hf {
       algorithm  HMAC_MD5;
       secret CLG2aS/NcdKWQpdJ9eULSw==;
};
```

主DNS服务器重启后在从DNS服务器上删除文件，并重启，此时应该无法获取到区域的副本。

然后在从DNS服务器上配置，指明密钥类型和密钥

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

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

server 192.168.24.244 {
        key { hf; };
};
key hf {
       algorithm  HMAC_MD5;
       secret CLG2aS/NcdKWQpdJ9eULSw==;
};
```

再次查看slaves目录，此时区域副本已经学习到。