实验名称	DNS 实验		
姓名		学号	

实验步骤

所有实验均在 VMware 17.6.1 虚拟机环境下的 Ubuntu 16.04.7 系统中完成。

一、Host1 设置

1、网卡编辑

2、安装 bind

```
ysn@ubuntu:~$ sudo apt-get install bind9
[sudo] password for ysn:
Reading package lists... Done
Building dependency tree
Reading state information... Done
bind9 is already the newest version (1:9.10.3.dfsg.P4-8ubuntu1.19).
0 upgraded, 0 newly installed, 0 to remove and 186 not upgraded.
```

3、修改 DNS 服务器

```
ysn@ubuntu:~$ sudo gedit /etc/resolv.conf

(gedit:2430): IBUS-WARNING **: The owner of /home/ysn/.config/ibus/bus is not ro
ot!

(gedit:2430): IBUS-WARNING **: Unable to connect to ibus: Unexpected lack of con
tent trying to read a line

(gedit:2430): Gtk-WARNING **: Calling Inhibit failed: GDBus.Error:org.freedeskto
p.DBus.Error.ServiceUnknown: The name org.gnome.SessionManager was not provided
by any .service files

** (gedit:2430): WARNING **: Set document metadata failed: Setting attribute met
adata::gedit-spell-enabled not supported

** (gedit:2430): WARNING **: Set document metadata failed: Setting attribute met
adata::gedit-encoding not supported
```

Open | resolv.conf /etc | Save

```
# Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolvconf(8)
#     DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
nameserver 127.0.0.1
search localdomain
```

4、在配置文件/etc/bind/named.conf.local 中添加新的域

```
ysn@ubuntu:/etc/bind$ sudo gedit named.conf.local

(gedit:5238): IBUS-WARNING **: The owner of /home/ysn/.config/ibus/bus is not ro
ot!
```

Open | *named.conf.local /etc/bind | Save

```
//
// Do any local configuration here
//

// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";

zone "yanshannan.com"{
        type master;
        file "/etc/bind/db.yanshannan.com";
        };
```

5、以 db.local 为模版设置自己的 DNS 映射

```
ysn@ubuntu:/etc/bind$ sudo cp db.local db.yanshannan.com
ysn@ubuntu:/etc/bind$ sudo gedit db.yanshannan.com

(gedit:5257): IBUS-WARNING **: The owner of /home/ysn/.config/ibus/bus is not ro
ot!
```

Open | db.yanshannan.com /etc/bind | Save

```
;
; BIND data file for local loopback interface
;
$TTL    604800
@       IN      SOA     yanshannan.com. mail.yanshannan.com. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      yanshannan.com.
@       IN      A       183.2.172.185
@       IN      AAAA    ::1
*       IN      A       183.2.172.185
```

6、启动 bind9

```
ysn@ubuntu:/etc/bind$ sudo /etc/init.d/bind9 start
[ ok ] Starting bind9 (via systemctl): bind9.service.
```

7、清空 DNS 缓存

```
ysn@ubuntu:/etc/bind$ sudo /etc/init.d/dns-clean start
Running 0dns-down to make sure resolv.conf is ok...done.
```

8、测试

```
ysn@ubuntu:/etc/bind$ sudo /etc/init.d/bind9 restart
[ ok ] Restarting bind9 (via systemctl): bind9.service.
ysn@ubuntu:/etc/bind$ sudo /etc/init.d/dns-clean start
Running 0dns-down to make sure resolv.conf is ok...done.
```

```
ysn@ubuntu:/etc/bind$ ping www.yanshannan.com
PING www.yanshannan.com (183.2.172.185) 56(84) bytes of data.
64 bytes from 183.2.172.185: icmp_seq=1 ttl=128 time=10.5 ms
64 bytes from 183.2.172.185: icmp_seq=2 ttl=128 time=11.2 ms
64 bytes from 183.2.172.185: icmp_seq=3 ttl=128 time=9.39 ms
64 bytes from 183.2.172.185: icmp_seq=4 ttl=128 time=8.66 ms
64 bytes from 183.2.172.185: icmp_seq=5 ttl=128 time=9.02 ms
64 bytes from 183.2.172.185: icmp_seq=6 ttl=128 time=8.79 ms
64 bytes from 183.2.172.185: icmp_seq=7 ttl=128 time=8.91 ms
64 bytes from 183.2.172.185: icmp_seq=8 ttl=128 time=8.85 ms
64 bytes from 183.2.172.185: icmp_seq=9 ttl=128 time=8.60 ms
64 bytes from 183.2.172.185: icmp_seq=10 ttl=128 time=7.67 ms
^C
--- www.yanshannan.com ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9016ms
rtt min/avg/max/mdev = 7.671/9.169/11.234/0.966 ms
```

数据分析

一、www.yanshannan.com 和其他域名的解析过程

在 DNS 服务中，www.yanshannan.com 与其他域名 X 的解析过程遵循以下基本步骤：

1. 客户端发起请求

在浏览器中输入 www.yanshannan.com 或 X 时，浏览器首先会检查本地缓存，看看是否有该域名的 IP 地址。如果没有缓存，浏览器会将解析请求发送到系统配置的 DNS 解析器。

2. DNS 解析器检查本地缓存

DNS 解析器首先检查其缓存中是否有 www.yanshannan.com 或 X 的解析记录。如果有且记录未过期，解析器会直接返回缓存中的 IP 地址（对应 www.yanshannan.com 的情况）。

3. 递归查询过程

如果 DNS 解析器没有缓存该域名的解析记录（对应 X 的情况），它会开始进行递归查询，具体步骤如下：

（1）根域名服务器

DNS 解析器向根域名服务器发起请求。根域名服务器的职责是提供顶级域（TLD）服务器的地址。

（2） TLD 服务器

DNS 解析器接着向.com 的 TLD 服务器发送请求。TLD 服务器会查询到 X 的权威 DNS 服务器地址。

（3）权威 DNS 服务器

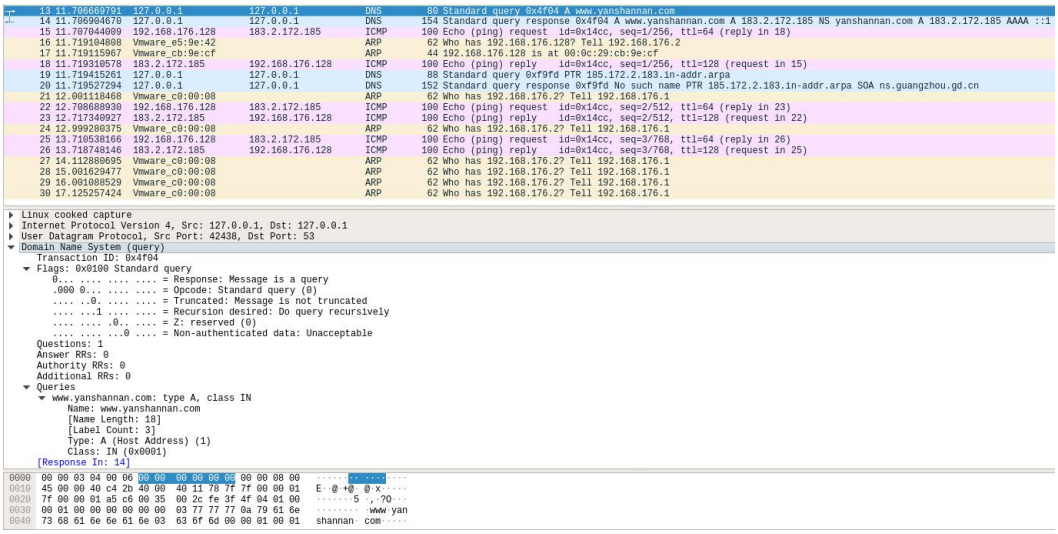
DNS 解析器接着向其它地址的权威 DNS 服务器发送查询请求，询问 X 对应的 IP 地址。权威 DNS 服务器会返回该域名的解析记录（通常是 A 记录或 AAAA 记录，分别对应 IPv4 和 IPv6 地址）。

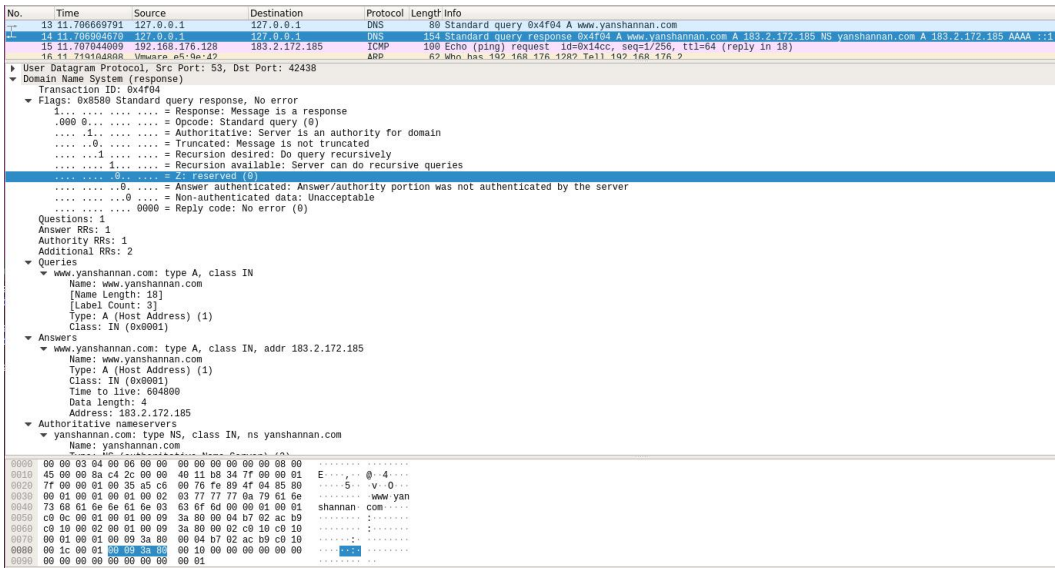
（4）返回解析结果

最后，DNS 解析器将从权威 DNS 服务器获得的 IP 地址返回给浏览器，浏览器就可以用该 IP 地址与服务器建立连接，加载网页。

www.yanshannan.com 地址解析过程的具体实例可以参考问题二中提供的截图和分析。

二、使用 wireshark 抓包分析域名请求和应答过程





可以看出，第 1 张图展示了域名请求过程：

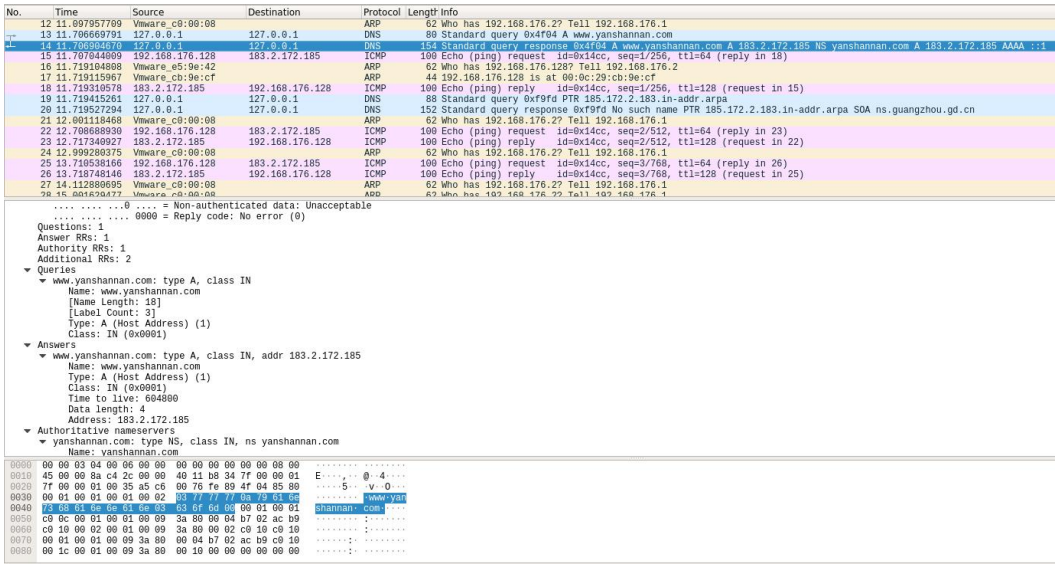
- 源 IP：127.0.0.1（本地回环地址）
- 目标 IP：127.0.0.1（本地 DNS 服务）
- 查询类型：A 记录查询，表示请求解析域名到 IPv4 地址。
- 域名：www.yanshannan.com
- 传输层协议：UDP，端口 53（DNS 通信端口）。

第 2 张图展示了域名应答过程：

- A 信息：返回 IPv4 地址 183.2.172.185。
- NS 记录：域名服务器为 yanshannan.com。
- AAAA 信息：返回 IPv6 地址 ::1，这是回环地址，表示当前系统没有真实的 IPv6 地址。

最终结果为：域名 www.yanshannan.com 成功解析为 183.2.172.185。

三、部分示例数据



思考题	**1、一个 DNS 查询的答复中是否会包含几个应答记录？如果是，对同一查询多执行几次，看看每次应答记录的顺序是否相同，试分析为什么。** 是的，一个 DNS 查询的答复可能包含多个应答记录。这种情况通常发生在以下几种情形： （1）如果查询的域名是一个 CNAME（Canonical Name，别名），那么应答会包含原始的查询记录以及 CNAME 指向的实际域名的解析结果。 （2）DNS 通常用于实现负载均衡。如果多个 IP 地址对应一个域名，应答记录会返回所有这些 IP 地址。 （3）某些类型的 DNS 查询可能返回附加记录，例如 MX（邮件交换）记录会附带相应邮件服务器的 A 记录或 AAAA 记录。 在多次执行同一查询的情况下，返回的应答记录顺序可能不相同。以下是原因分析： （1）许多 DNS 服务器会随机化返回的记录顺序，以实现负载均衡。例如，如果一个域名解析到多个 IP 地址，DNS 服务器会随机改变它们的排列顺序，以平衡用户请求到不同的服务器。 （2）一些中间的 DNS 缓存服务器可能根据策略调整记录顺序，例如优先返回延迟较低的记录，或者将最近最频繁访问的记录排在前面。 （3）根据 DNS 协议规范（RFC 1034 和 RFC 1035），一个资源记录集（RRset，即同一域名、同一类型的记录）中的记录是等价的，客户端应能够正确处理任意排列顺序。 **2、思考一下如何劫持 www.naichabiao.com 到 www.jd.com.** DNS 劫持是一种攻击技术，通过篡改 DNS 查询过程中的数据，将用户引导到非目标网站。以下是关于这种劫持的四种可能实现方式的分析。 （1）修改目标用户设备上的 hosts 文件，使 www.naichabiao.com 的解析指向 www.jd.com 的 IP 地址。 （2）向目标用户的 DNS 服务器发送伪造响应，将 www.naichabiao.com 的解析指向 www.jd.com。具体来说，伪造响应数据包，构造一个包含 www.naichabiao.com 的查询，附加 www.jd.com 的 IP，确保数据包的源地址与真实 DNS 服务器一致。 （3）拦截目标用户和 DNS 服务器之间的通信，修改 www.naichabiao.com 的解析结果。具体来说，利用网络攻击工具（如 ARP 欺骗）劫持用户到 DNS 服务器的流量。替换响应中的真实记录，将 www.naichabiao.com 的解析结果修改为 www.jd.com 的 IP。 （4）如果攻击者能够控制目标用户使用的公共 DNS 服务，则可以直接篡改解析记录。具体来说，登录或渗透目标公共 DNS 服务器。将 www.naichabiao.com 的解析记录替换为 www.jd.com 的 IP 地址。
经验总结	*（实验过程中遇到的困难，试验中需要额外注意的事项，实验中激发的灵感等）* 1、在使用 ping 进行测试前，需要先重新启动 bind9 服务； 2、实验指导书中给的地址 180.97.33.108 ping 不通，换了百度的地址 183.2.172.185 才好； 3、wireshark 抓包时，看网卡 ens33 是看不到 DNS 服务的，必须去 any 看。