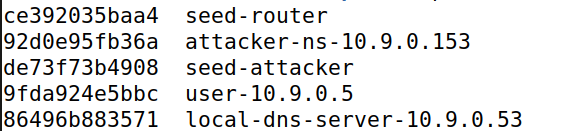
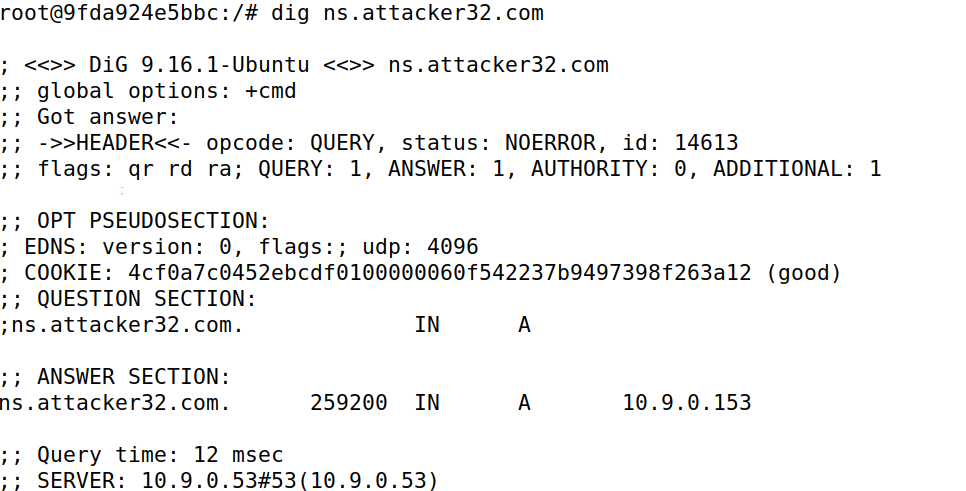
Lab 5 57118226 舒钰淇

各主机IP

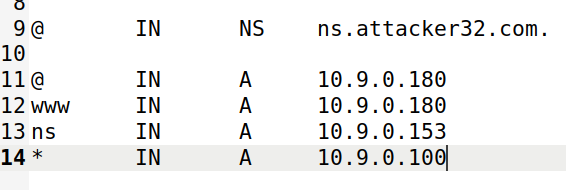


Task0: Testing the DNS Setup

先在host上dig ns.attacker32.com 解析到的地址与攻击者主机拥有的域名地址相同

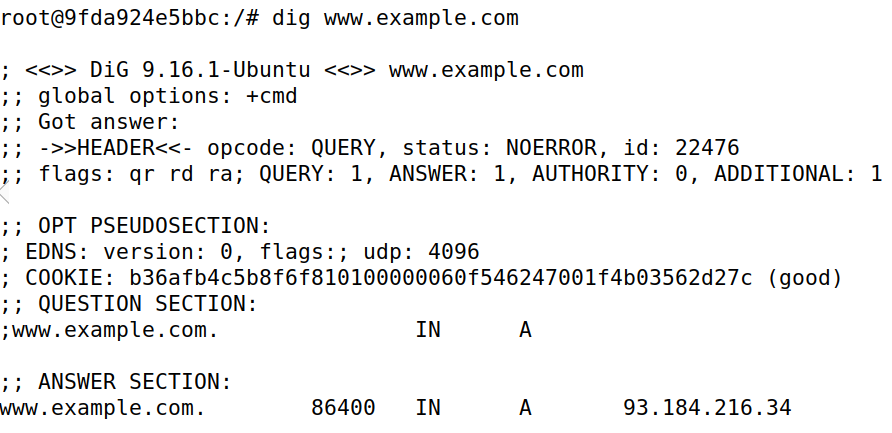


host上执行dig ns.attacker32.com

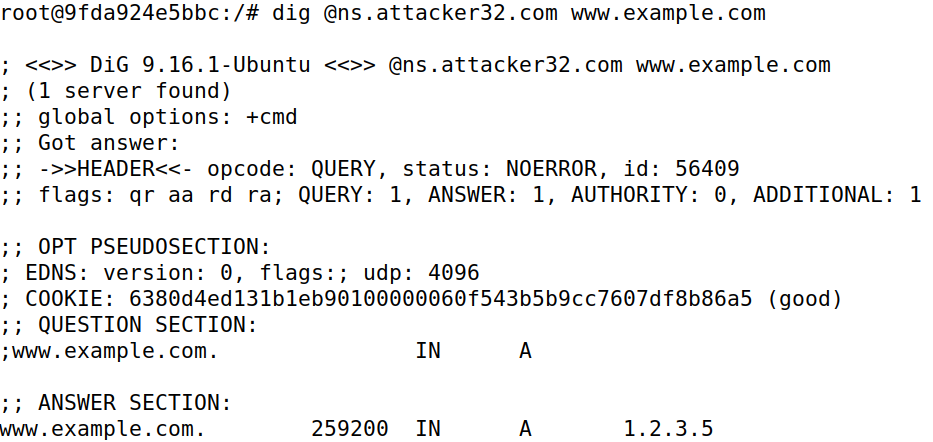


攻击者主机中的域文件内容

直接在host上dig www.example.com，得到结果

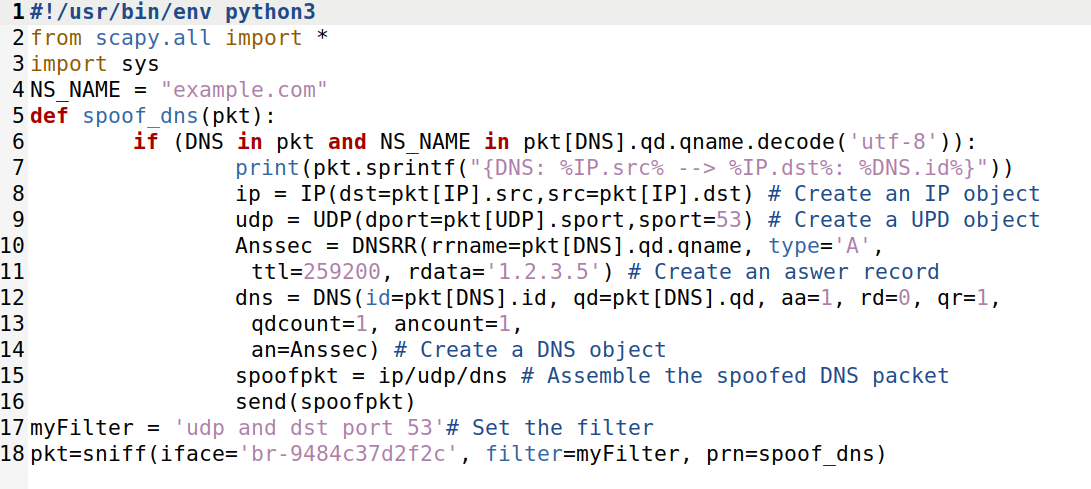


接着使用 dig @ns.attacker32.com www.example.com，向攻击者主机询问地址，得到相应的结果，由于攻击者主机也有相关域名解析信息，所以得到与上面不同的结果

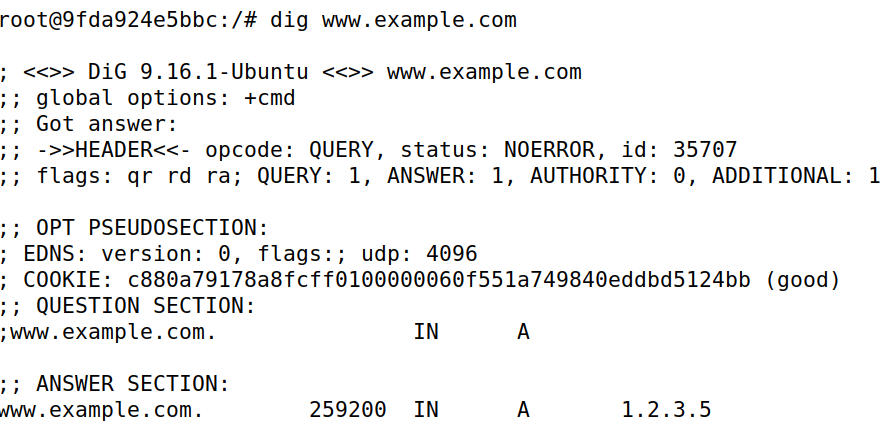


Task 1: Directly Spoofing Response to User

编写如下DNS嗅探和欺骗程序，在攻击者主机上运行，当收到对example.com的解析请求时，返回DNS答复报文

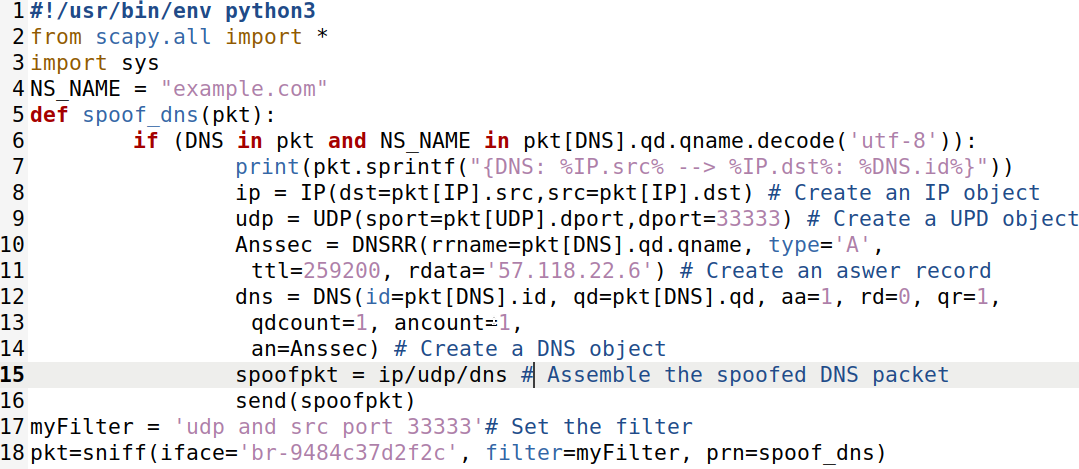


在本地DNS服务器上清空缓存后，在host上dig www.example.com，可以看到攻击成功。

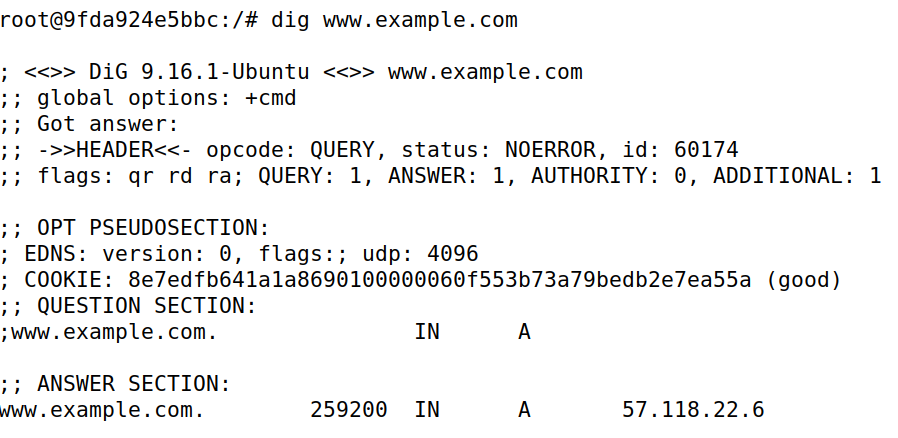


Task 2: DNS Cache Poisoning Attack – Spoofing Answers

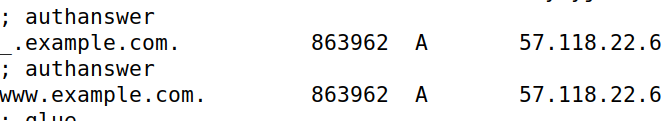
对Task1的程序进行修改，因为本地DNS服务器在收到未知的DNS请求时，需要由33333端口向外发送DNS请求报文进行查询，所以对相应的源宿端口进行修改，当收到example.com的DNS解析请求时，攻击者发送一个伪造答复报文将IP地址解析为57.118.22.6（我的学号）。



在host上dig www.example.com,可以看到攻击成功。

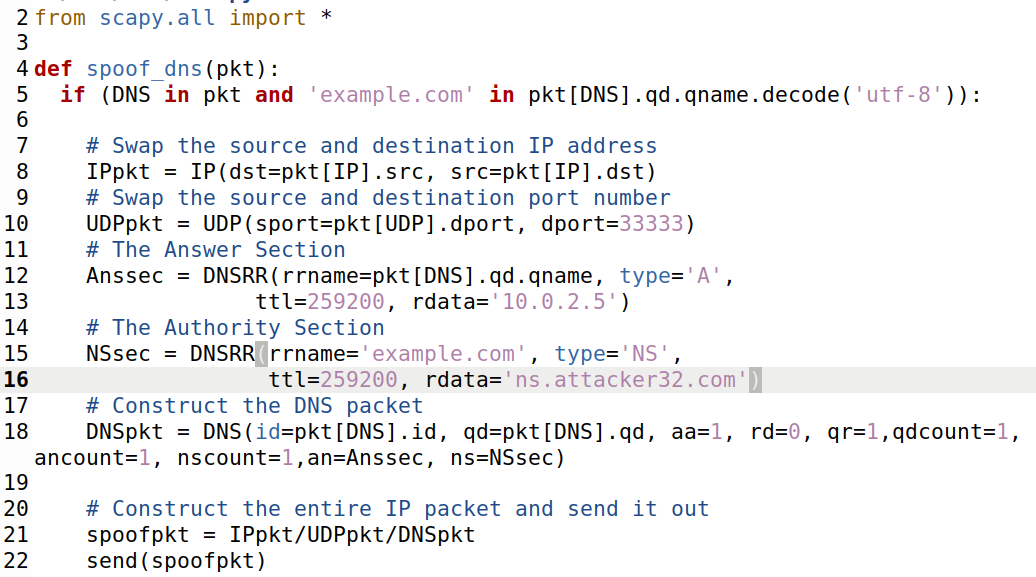


在本地DNS服务器上查看缓存信息，可以看到成功将相应的DNS解析信息写入缓存中。

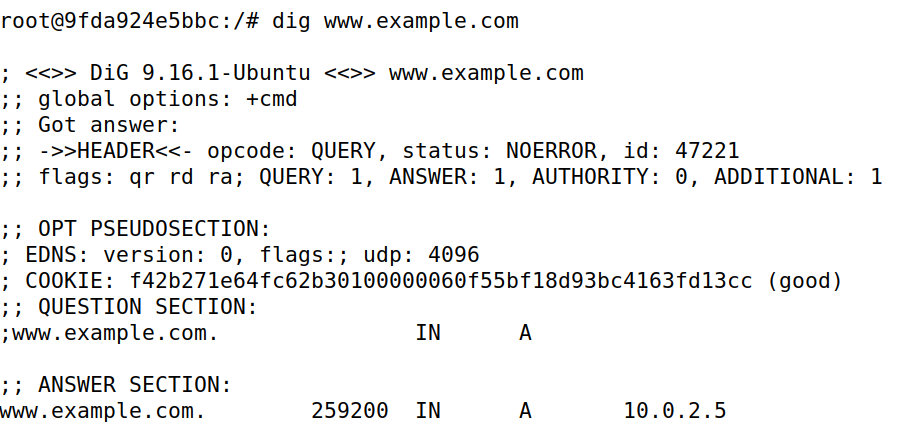


Task 3: Spoofing NS Records

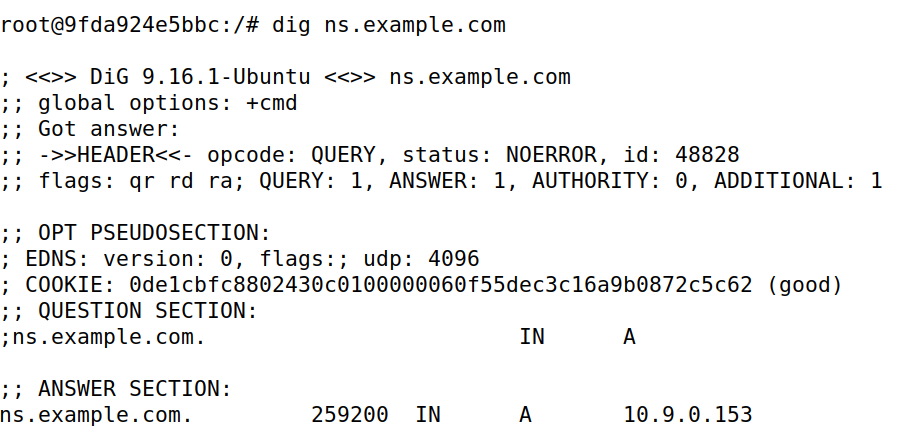
将代码进行如下修改，将域example.com的权威域名服务器改为ns.attacker32.com

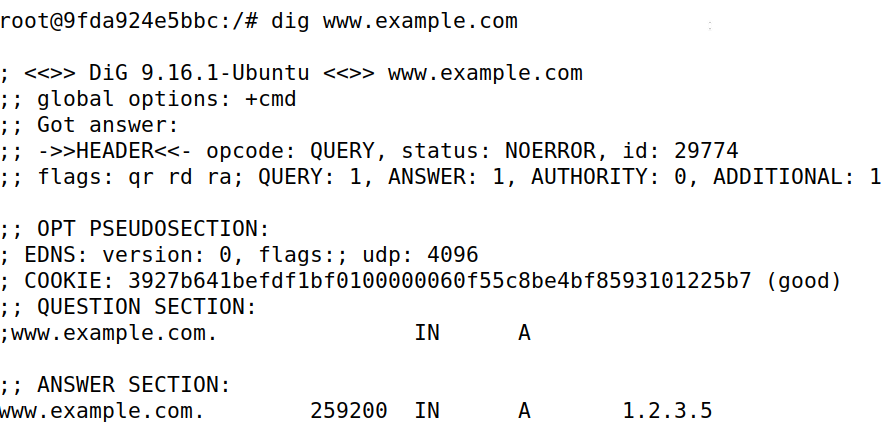


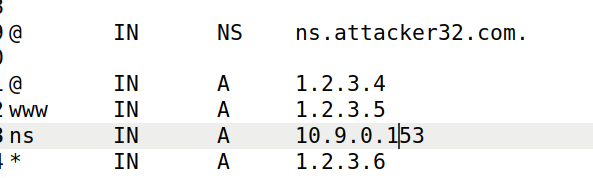
先在host 上dig www.example.com



接下来测试是否改变权威域名服务器，停止运行程序，在host上执行dig命令，可以看到DNS解析信息与攻击者上的域文件信息相同

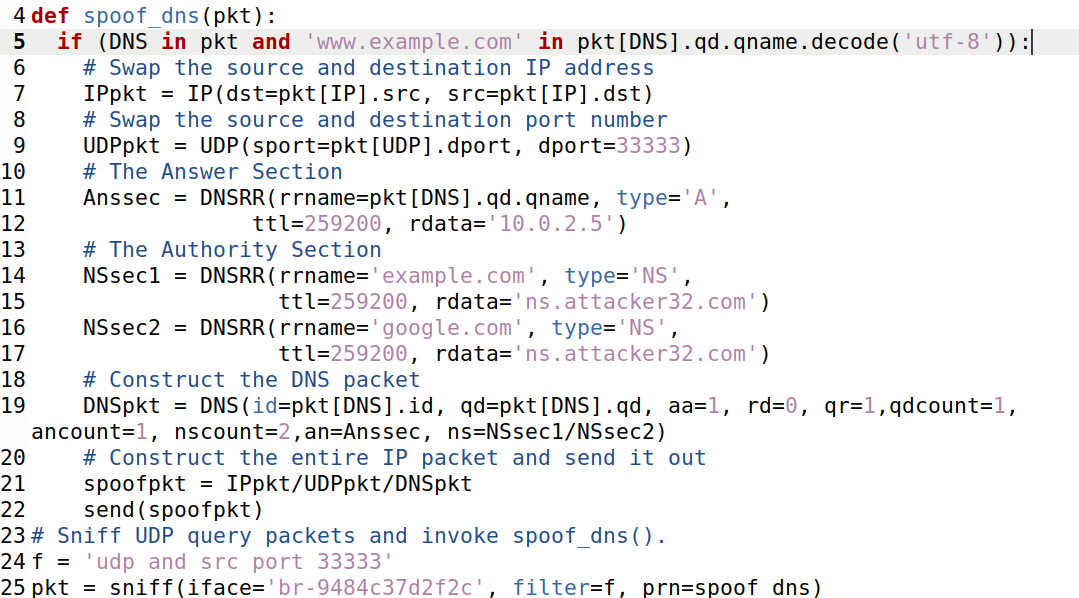






Task 4: Spoofing NS Records for Another Domain

修改代码如下图所示，在权威域名服务器内容中加入对google.com的权威域名服务器部分。

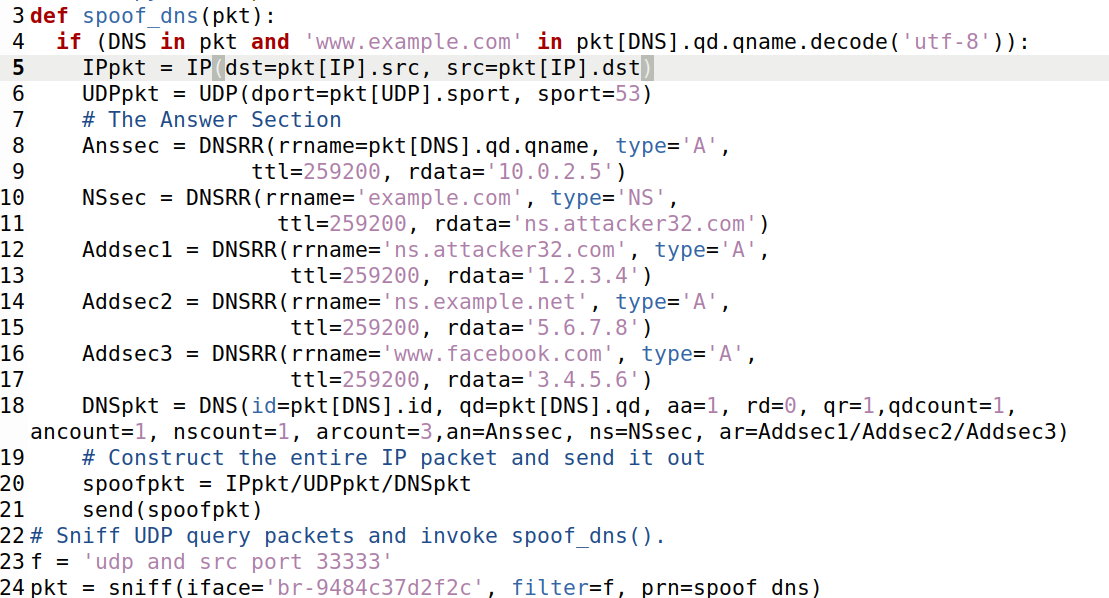


清空本地DNS服务器缓存后执行攻击，查看DNS服务器缓存，发现没有google.com的权威域名服务器，因为若成功则可以使未知权威域名服务器掌管任意域，不安全，所以该部分被丢弃。

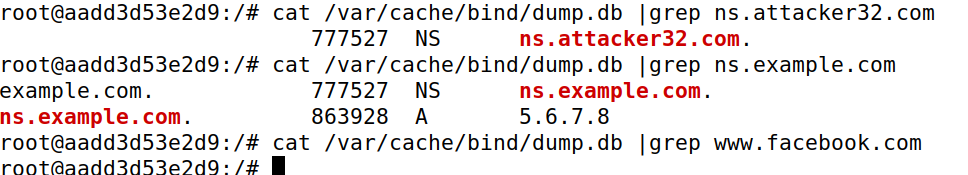
‘’

Task 5: Spoofing Records in the Additional Section

将程序进行如下修改，添加3条



在host主机上dig www.example.com后查看本地DNS服务器的缓存信息，可以看到存在ns.attacker32.com和ns.example.com的信息，而不存在www.facebook.com的信息，说明DNS服务器会接受附加部分中与权威服务器有关的信息，而如果附加部分是无关的信息则丢弃。因此我们可以将想要DNS劫持的域名放在权威域名服务器中，再在附加部分加入相关信息来达到攻击目的。



Additional Task 卡明斯基攻击

（由于该实验和截图是我之前做的，所以没有用现在实验环境中docker）

一．远程与本地DNS攻击的分析

本地DNS缓存中毒攻击局限性：

为了嗅探受害者的DNS请求攻击者和受害者的计算机必须在同一个局域网中。

远程攻击障碍：

1. 不能嗅探到受害者发出的DNS请求，需要对UDP报文中16比特的端口号和16比特的交易ID进行猜测。

2. 考虑到缓存影响，需要在受害者收到正确答复时将具有正确端口号和交易ID的伪造答复发送到受害者主机。

二．远程DNS攻击需要完成的任务

1.发送DNS请求：

向受害者DNS服务器发送想要掌管的域中某个网址的DNS请求。

2.发送欺骗回复：

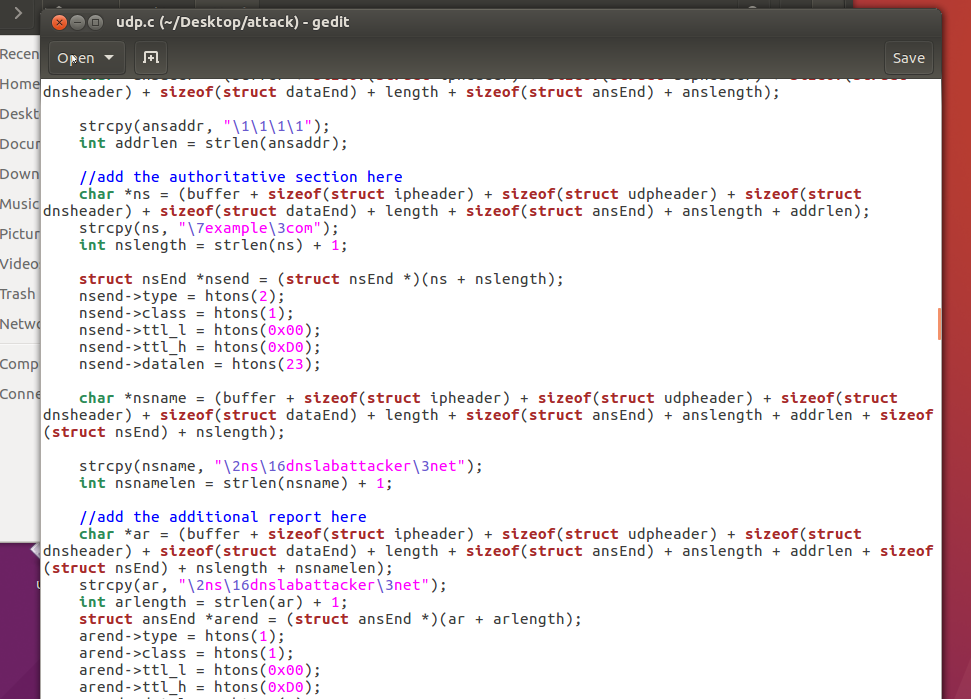
在受害者DNS服务器从根域名服务器等地收到正确答复之前，随机生成交易ID（端口号已知）的DNS答复报文，将攻击者主机指定为该域的权威服务器。

3.使缓存失效：

每次随机生成DNS请求报文中网址的名称，例如twysw.example.com，使得本地DNS服务器的缓存起不到作用。

三、攻击流程

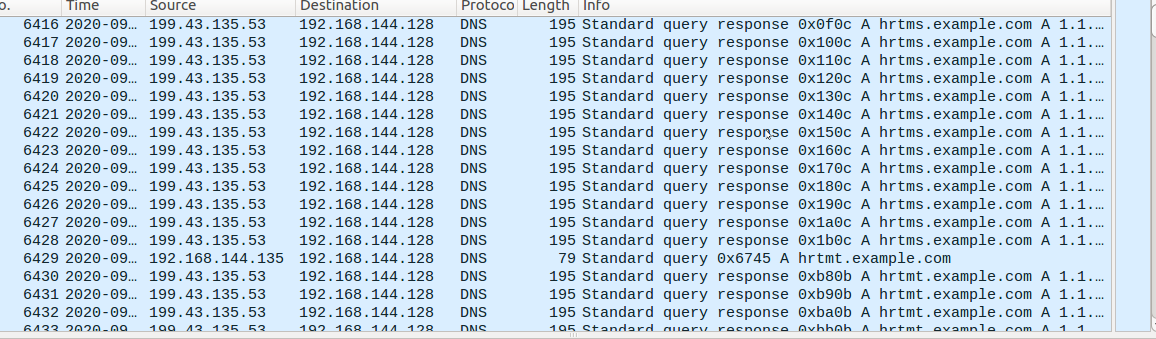
编写如下图所示C程序，先对本地DNS服务器发出xxxx.example.com的DNS请求（其中xxxx为随机生成的字符，在本地DNS服务器收到根域名服务器的正确答复之前向其发送大量DNS答复报文，每个报文随机生成交易ID，如果某个报文交易ID与DNS请求报文相匹配，则将使恶意DNS服务器成为example.com的权威域名服务器。若在收到正确答复前没有匹配的伪造答复，则更换DNS请求中的主机名再次进行攻击

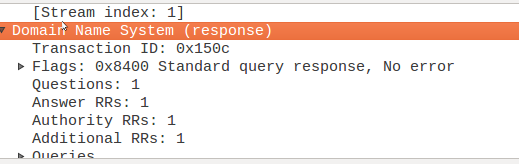


在本地DNS服务器上打开wireshark,，可以看到收到了大量DNS报文。其中的192.168.144.135是攻击者IP，192.168.144.128是DNS服务器IP，199.43.135.53为根域名服务器IP。

在发出大量伪造的hrtms.example.com的答复后，DNS服务器收到了正确的信息，因此重新开始攻击，向DNS服务器发出的是hrtmt.example.com的DNS请求。其中的每个请求用的都是不同的交易ID进行猜测。

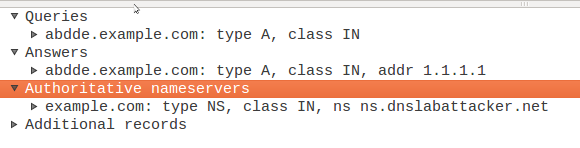
每个伪造DNS答复中的内容都是试图让攻击者的ns.dnslabattacker.net来掌管example.com的域



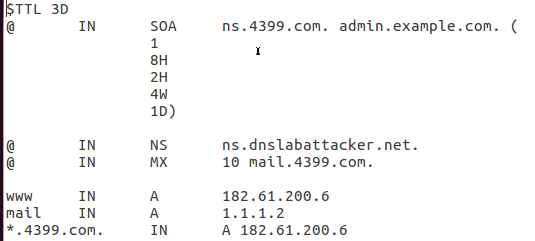


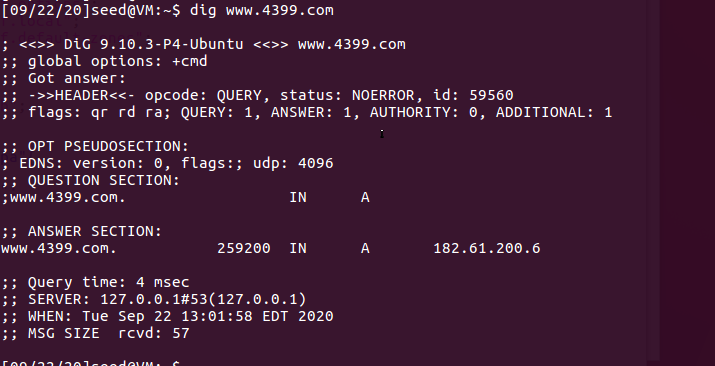






使用这种方法，可以将本地DNS服务器中4399.com.的域的掌管者变为攻击者主机。将www.4399.com的ip地址改为182.61.200.6（百度首页的IP地址）,两张图分别为攻击者DNS服务器上的4399.com的域的内容和在本地DNS服务器上dig www.4399.com的结果





在浏览器地址栏输入www.4399.com可以看到确实是向我们在攻击者DNS服务器上设置的IP发送TCP连接，但是由于保护机制，可以在第64号报文看到百度的服务器发出了拒绝请求的信息

