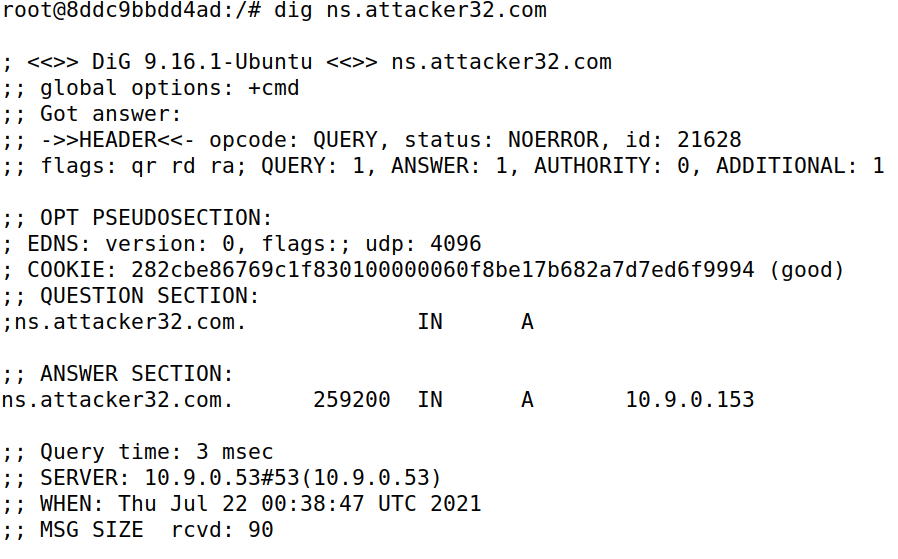
**Local DNS Attack Lab**

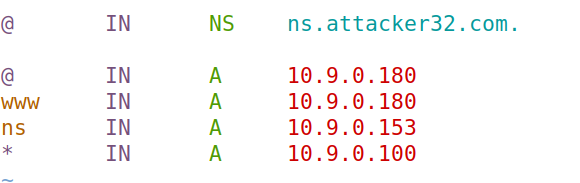
57118112 王怡乐

**Testing the DNS Setup**

**Get the IP address of ns.attacker32.com.**

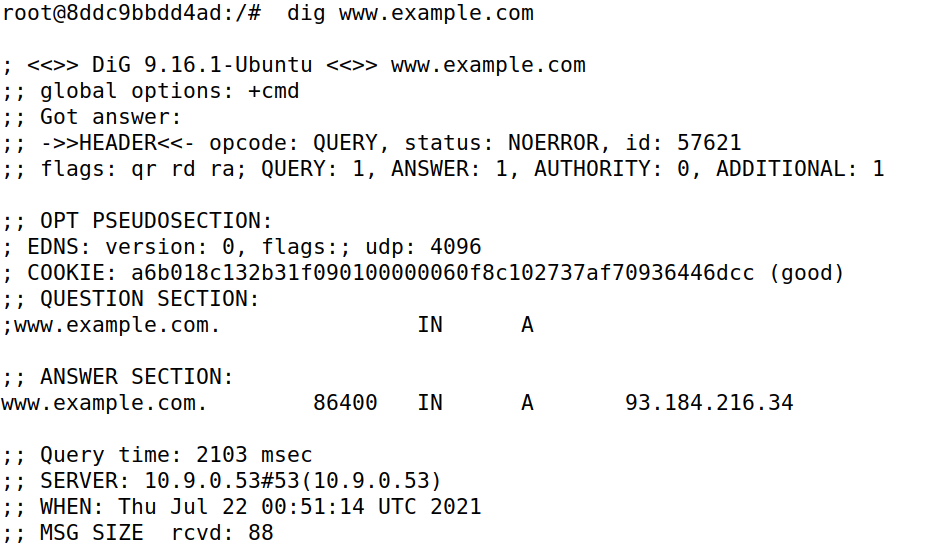


结果显示该域名指向的ip地址为10.9.0.153。查看攻击者域名服务器设置文件zone\_attacker32.com，发现结果显示的ip地址与文件中一致，设置无问题

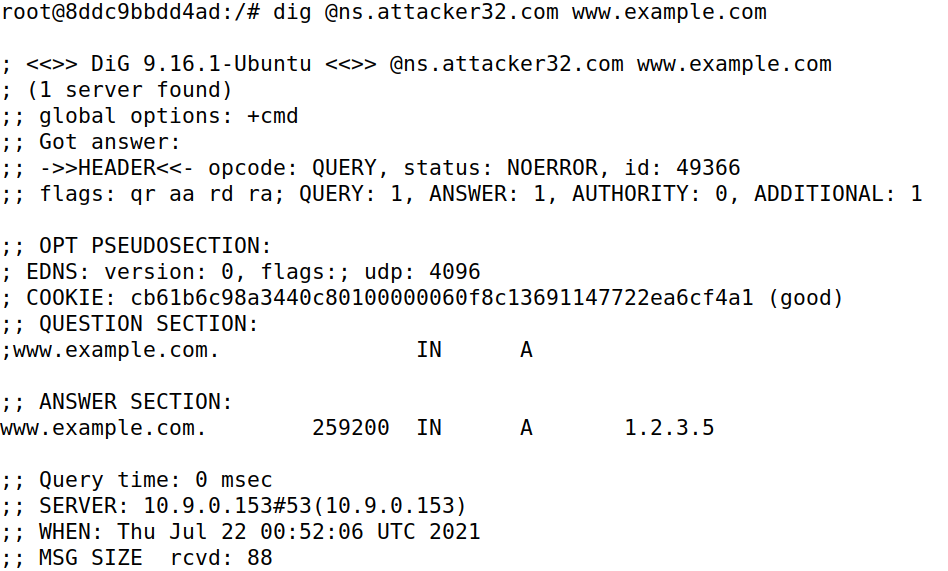


**Get the IP address of www.example.com**

执行 dig www.example.com 命令，输出结果如下：



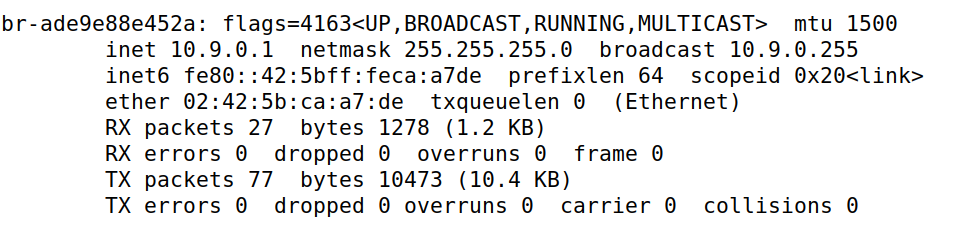
执行 dig @ns.attacker32.com www.example.com 命令，输出结果如下



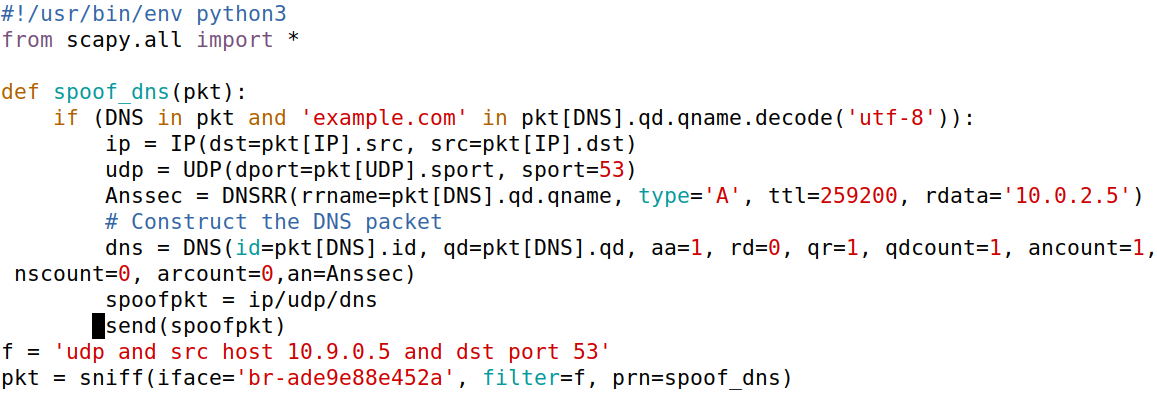
两个命令得到的 ip 地址不同，第一个命令直接从官方域名服务器获取信息，而第二个是从攻击者得到了假的结果

**Task 1: Directly Spoofing Response to User**

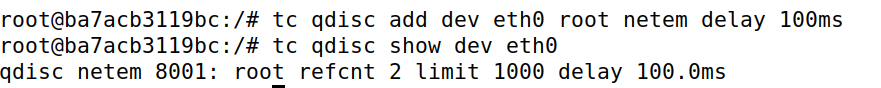
首先在攻击主机上查看 10.9.0.0/24 网段的端口名称



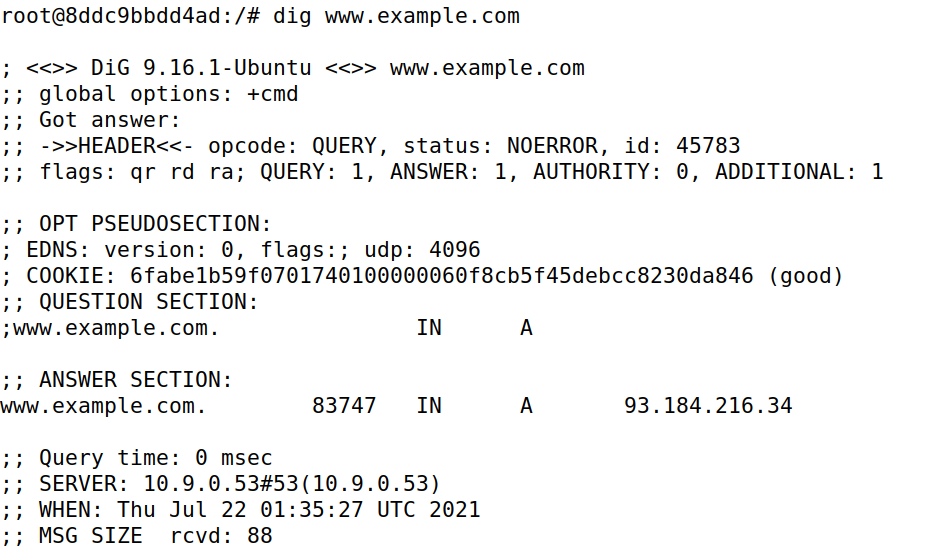
攻击代码如下。代码的过滤器只捕获源 ip 地址为用户主机，目的端口为 53 的 udp报文，即用户发送给域名服务器的 DNS 请求



在路由器上增加输出网络流量的延迟



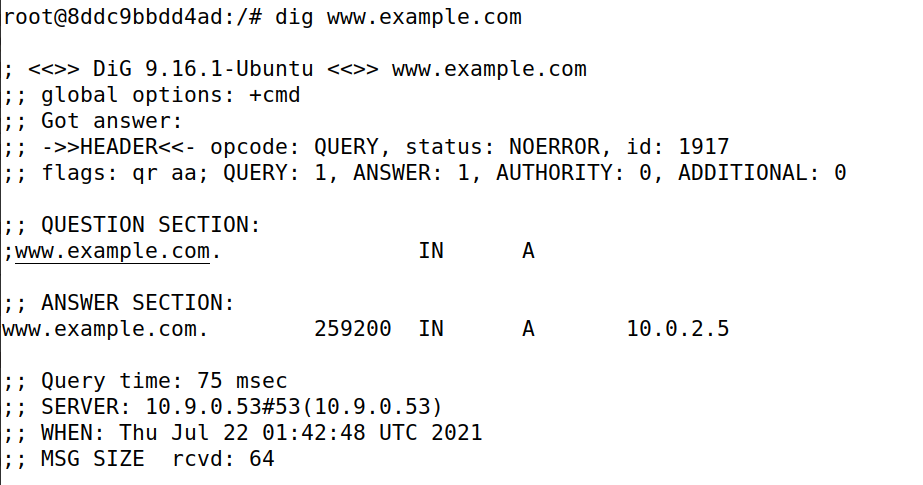
攻击前，在用户主机查询 www.example.com 的 DNS 信息



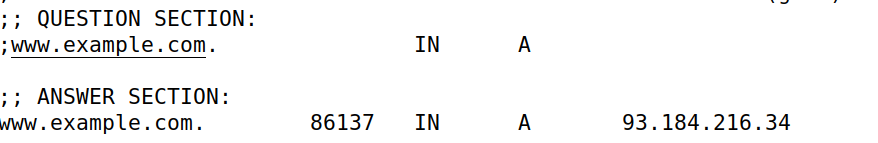
在本地域名服务器上清除缓存



执行攻击程序，再在用户主机查询 www.example.com 的 DNS 信息，发现 [www.example.com对应的ip地址变成了攻击者伪造的ip](http://www.example.com对应的ip地址变成了攻击者伪造的ip)地址，攻击成功



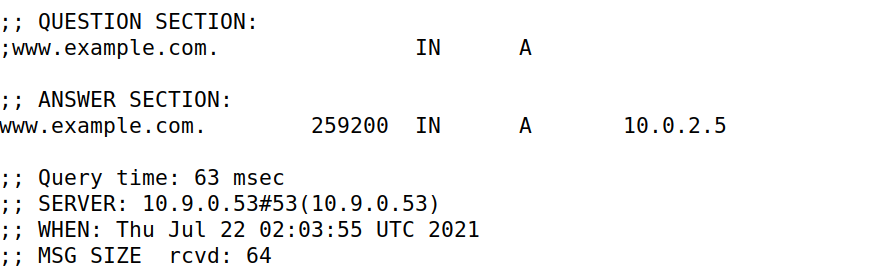
停止攻击，在用户主机查询 www.example.com 的 DNS 信息，发现www.example.com对应的ip地址恢复



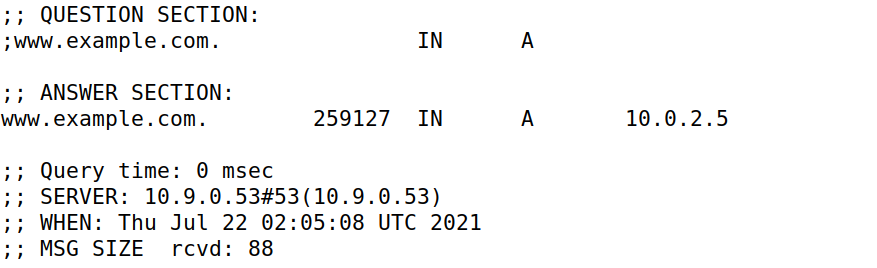
**Task 2: DNS Cache Poisoning Attack – Spoofing Answers**

修改攻击代码中的过滤器设置，将其更改为捕获本地域名服务器发往其他域名服务器的udp报文，攻击代码如下

攻击前在本地域名服务器上清除缓存。执行攻击时，DNS查询结果如下



停止攻击，DNS查询结果如下，发现攻击仍然成功



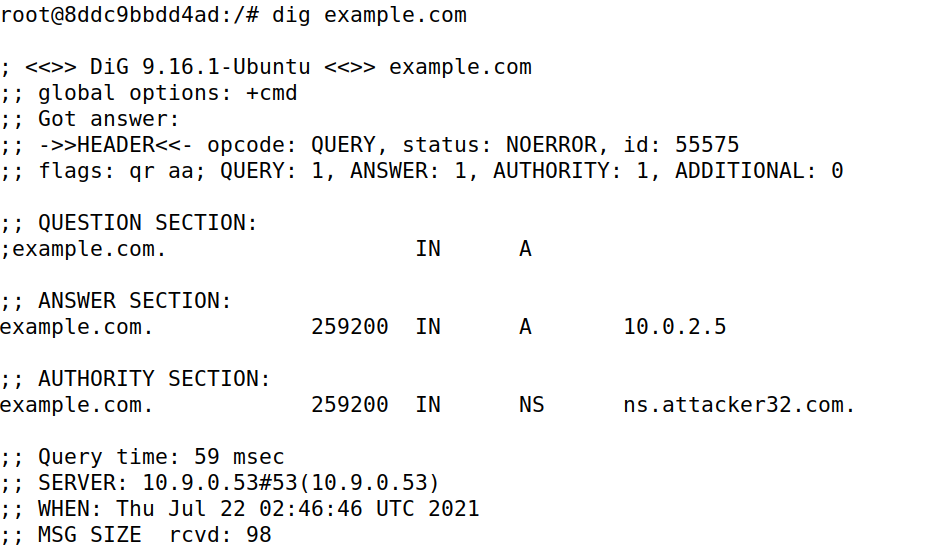
查看本地域名服务器的缓存，发现伪造的dns信息存在，攻击成功



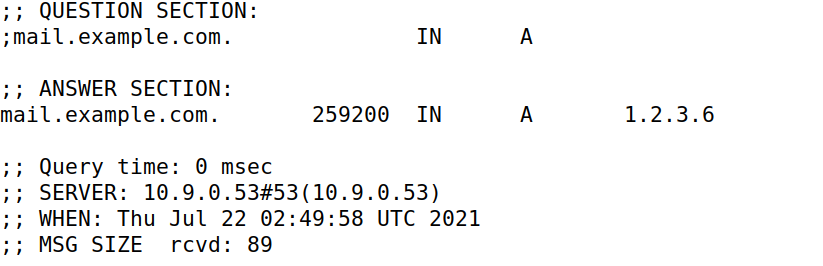
**Task 3: Spoofing NS Records**

修改攻击代码，增加一条NS记录，使ns.attacker32.com作为域名服务器，以便查询example.com域中的所有主机名，攻击代码如下

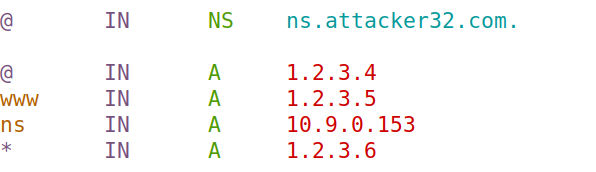
在本地服务器上清除缓存，然后执行攻击程序，在用户主机查询 www.example.com 的 DNS 信息，结果如下



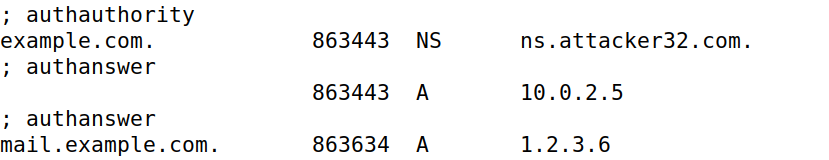
停止攻击，查询同一域名不同主机名的dns信息，结果显示该主机名指向ip地址1.2.3.6



查询文件zone.example.com，发现缺省时指向的地址为1.2.3.6，与上述结果一致，攻击成功

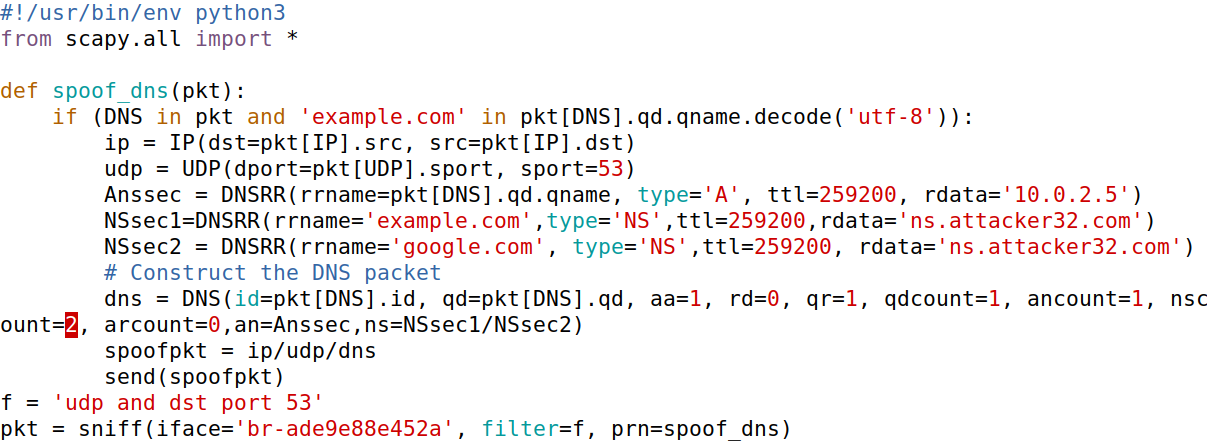


查看dns缓存，NS记录在缓存中

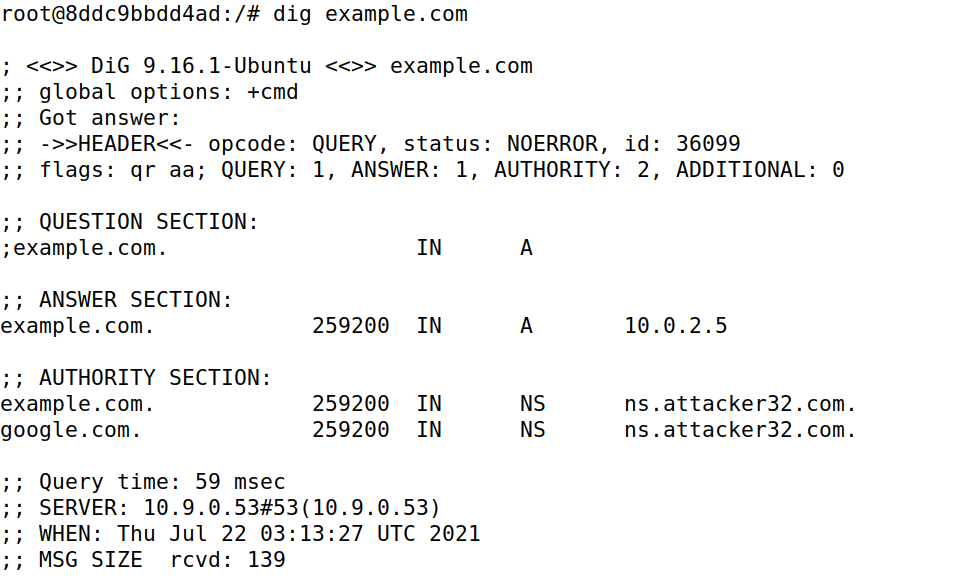


**Task 4: Spoofing NS Records for Another Domain**

修改代码，增加一条NS记录，使得google.com域名下的地址都指向ns.attacker32.com



在本地服务器上清除缓存，然后执行攻击程序，在用户主机查询 www.example.com 的 DNS 信息，结果显示有两条权威字段记录



查看缓存，发现只有example.com的NS记录

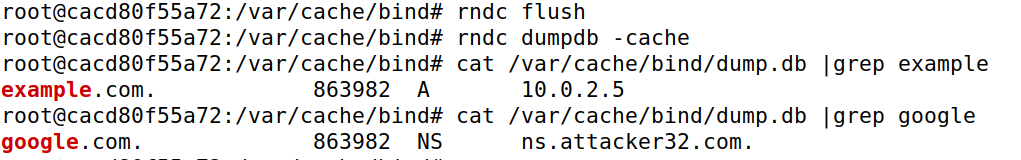




把代码中的两条NS记录交换顺序



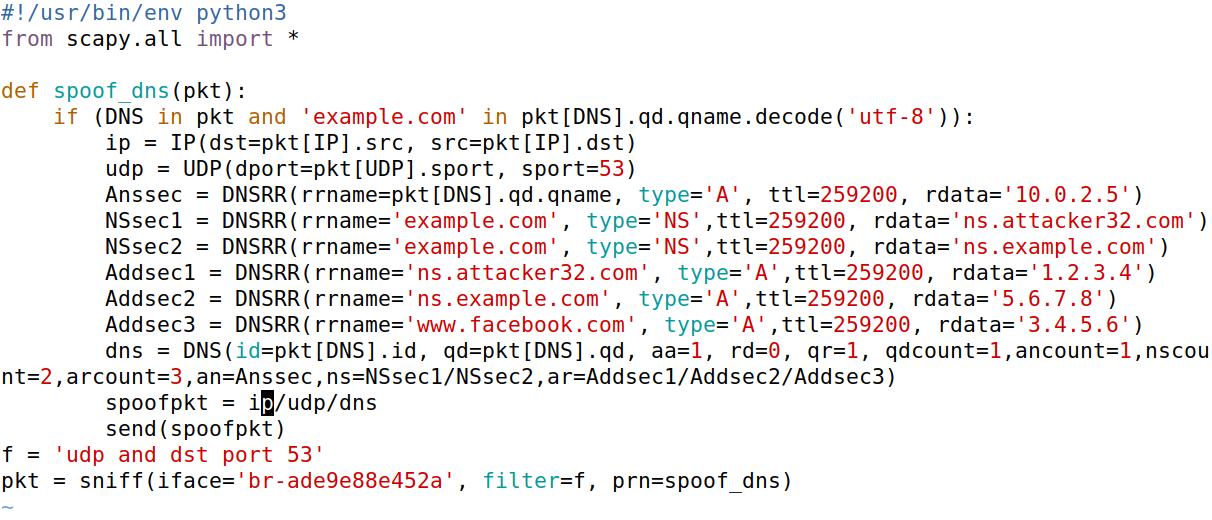
再次执行攻击，查看缓存，发现只有google.com的NS记录



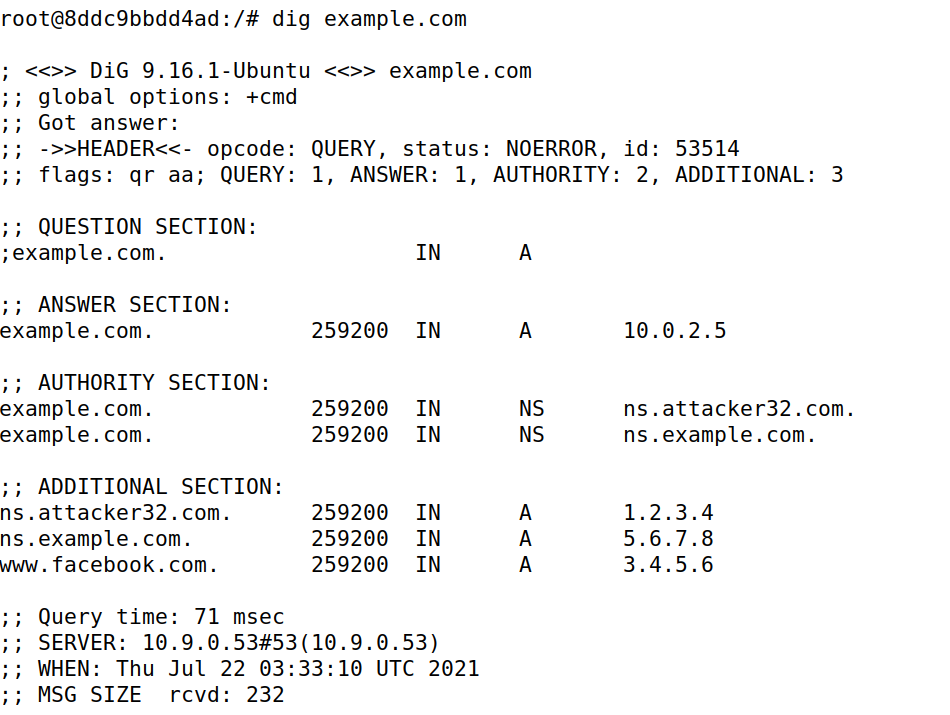
猜测可能只缓存第一条NS记录

**Task 5: Spoofing Records in the Additional Section**

修改代码，添加三条附加字段



清除缓存，执行攻击程序，结果如下



查看缓存，发现没有[www.facebook.com](http://www.facebook.com)的记录，这是由于附加字段 additional 中的记录只有与权威字段 authority 中条目相关，才会将其存入到 dns 的缓存中。

