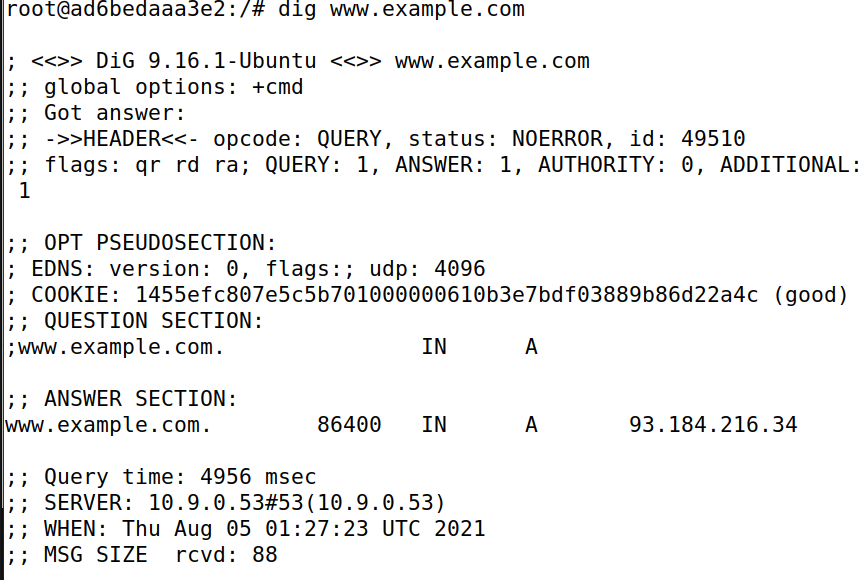
Lab 5

Testing the DNS Setup

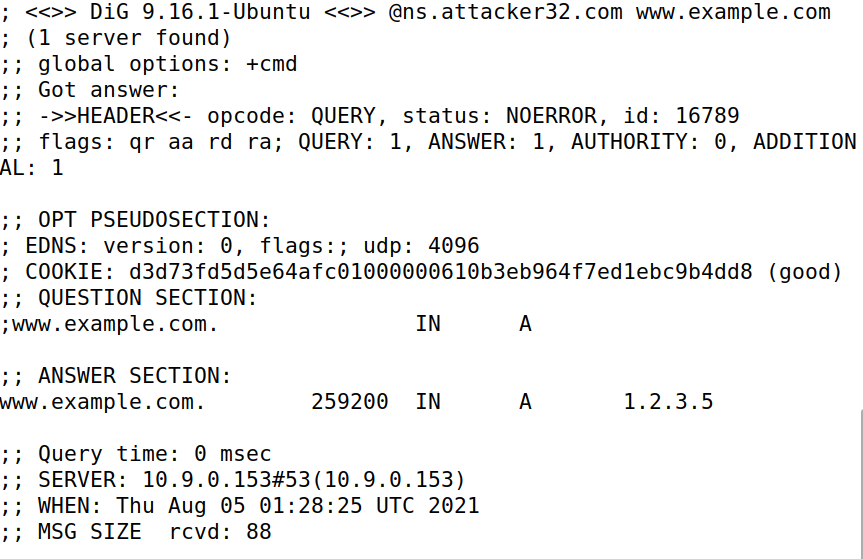
先在user上 dig ns.attacker32.com



直接 dig [www.example.com , 结果如下](http://www.example.com,结果如下)



通过attacker查询www.example.com，从攻击者那得到虚假结果

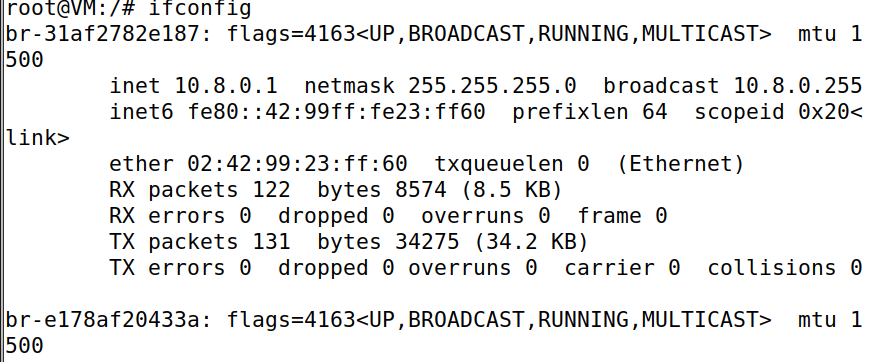


Task 1:

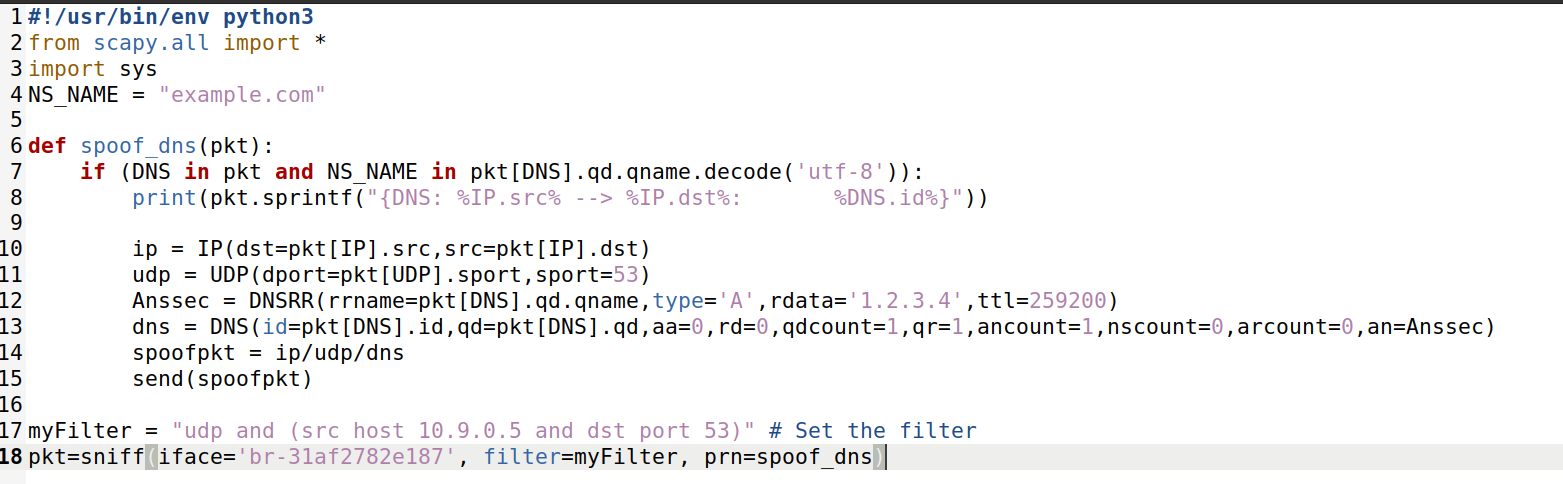
task1 的目的是捕获用户发出的 DNS 请求，然后返回一个假的 DNS 响应，只

要伪造的DNS响应在真的DNS响应到达用户主机前到达，用户就会接受伪造信息。

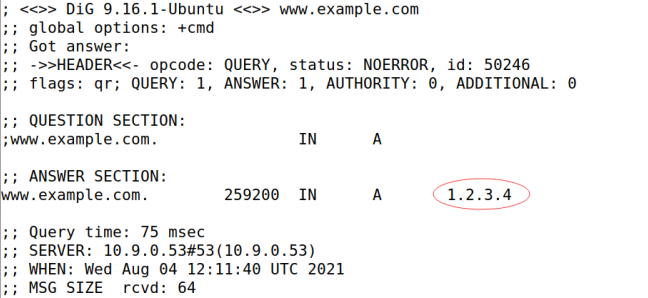
首先我们在攻击主机上查看 10.9.0.0/24 网段的端口名称，补充代码时会用到：



修改代码如下

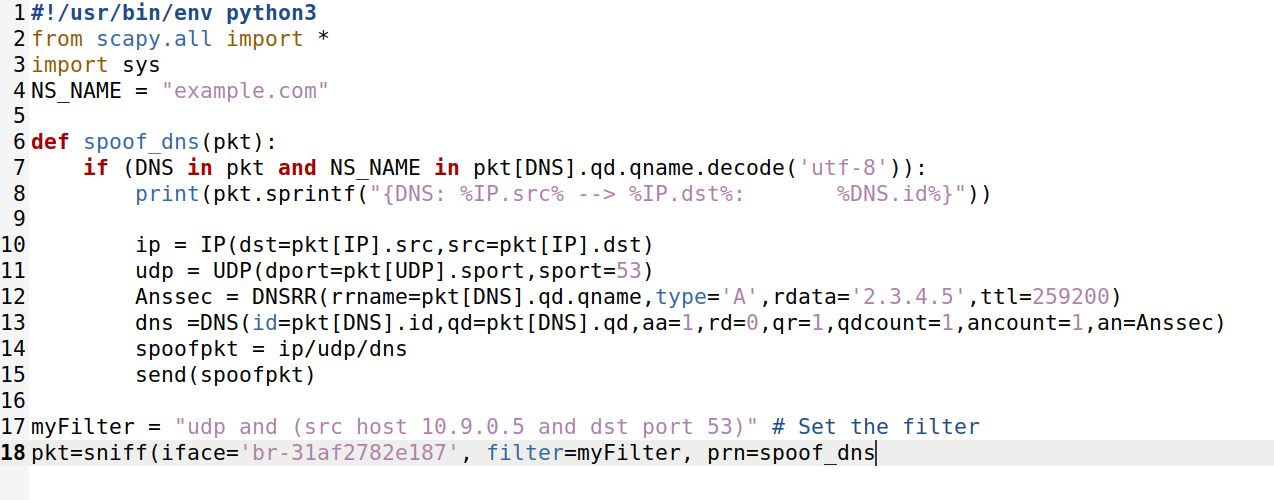


运行程序，再次查询dig [www.example.com发现报文被修改](http://www.example.com发现报文被修改)



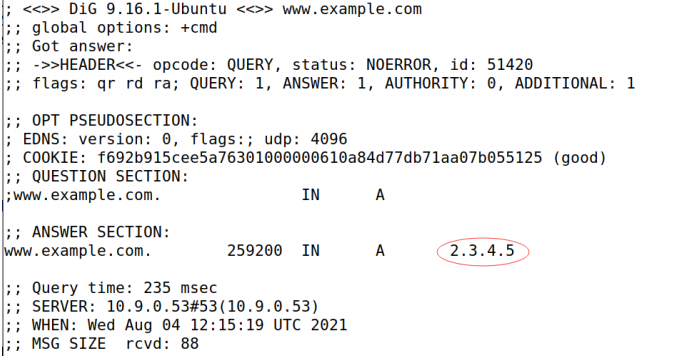
Task 2:

将sniff的目标IP改为DNS服务器的IP，修改代码如下

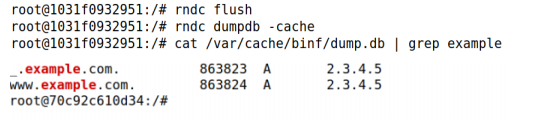


在运行攻击程序前，先刷新本地DNS服务器缓存rndc flush，然后dig [www.example.com](http://www.example.com) 结果与未攻击前一样

Attacker上运行攻击代码 查询 可以看到user被欺骗

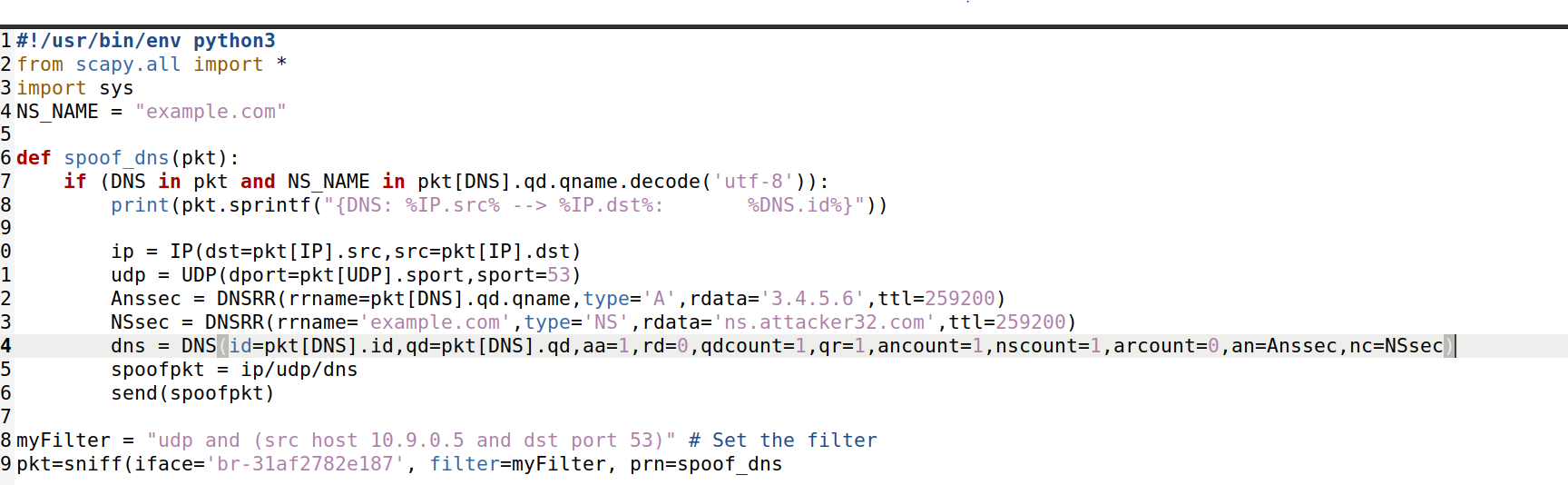


此时在本地dns服务器上运行rndc dumpdb -cache， cat/var/cache/bind/dump.db | grep example 可以看到缓存中毒攻击成功

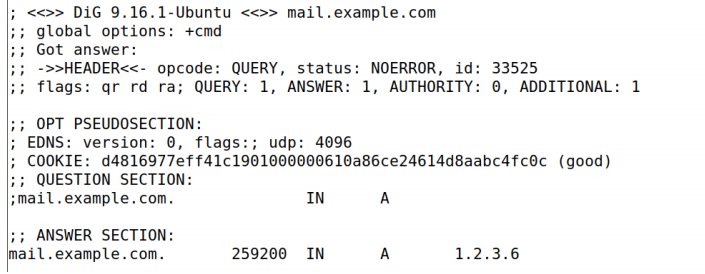
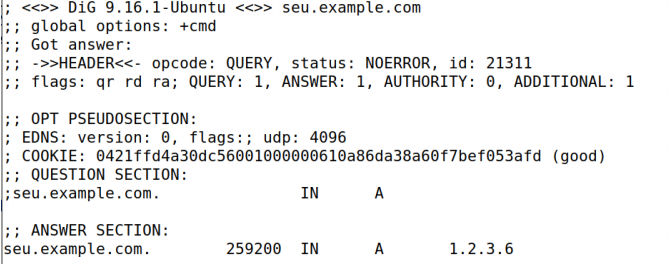
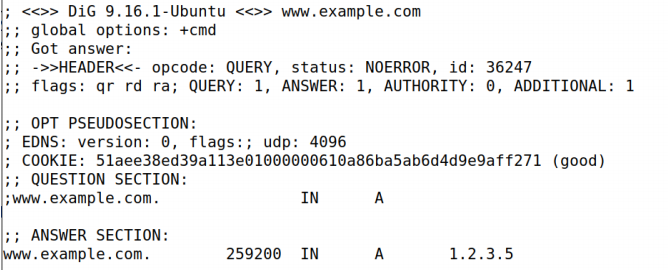


Task3 ：

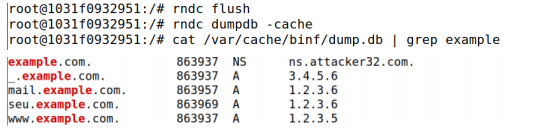
修改代码如下



运行攻击程序后，在user运行dig [www.example.com，](http://www.example.com，) dig seu.example.com dig mail.example,com可以看到均被欺骗



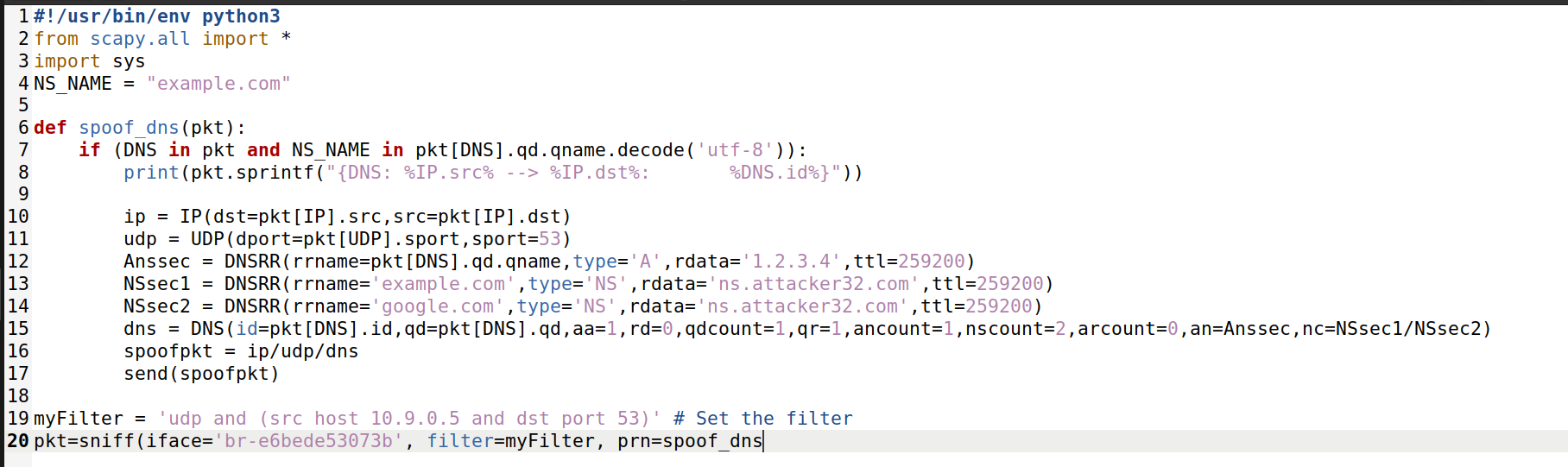
再来查看DNS服务器的cache 发现一家被污染为攻击者的路由器了



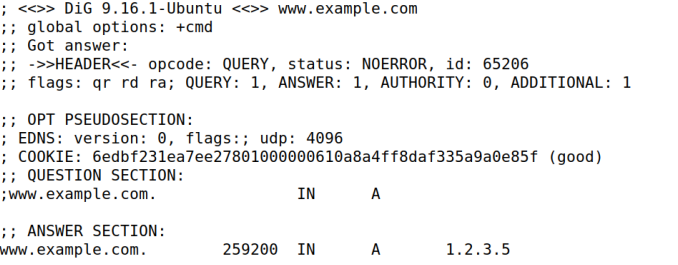
Task4：

在task3的基础上 增加一个新的ns记录，尝试google.com的nameserver也污染为攻击者的指令

代码修改如下



查询example.com可以发现攻击成功



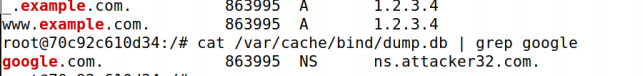
查看dns缓存

发现只有example.com的映射关系被写入缓存，而google.com并没有

于是将dns代码改为

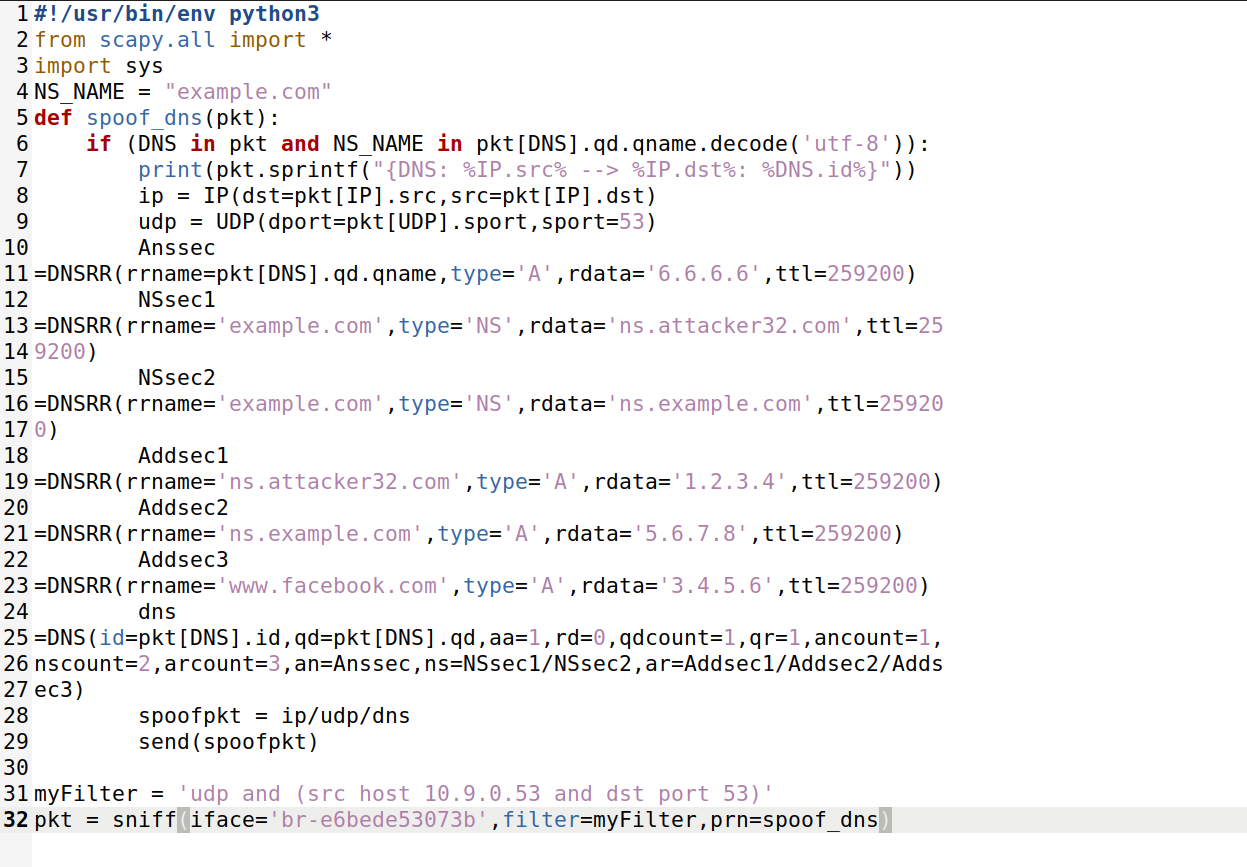
dns =DNS(...,ns=NSsec2/NSsec1)

重新进行攻击 发现Google.com到attacker.com的映射被存储到缓存中



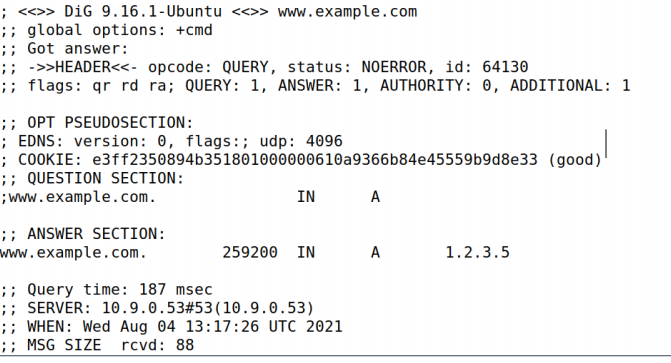
Task5：

修改代码如下



user查询www.example.com，发现攻击成功，但是发挥作用的是ns的伪造报文，而非写在

响应里的6.6.6.6



查看dns缓存

