**lab[5]-report**

——57118207魏宇萌

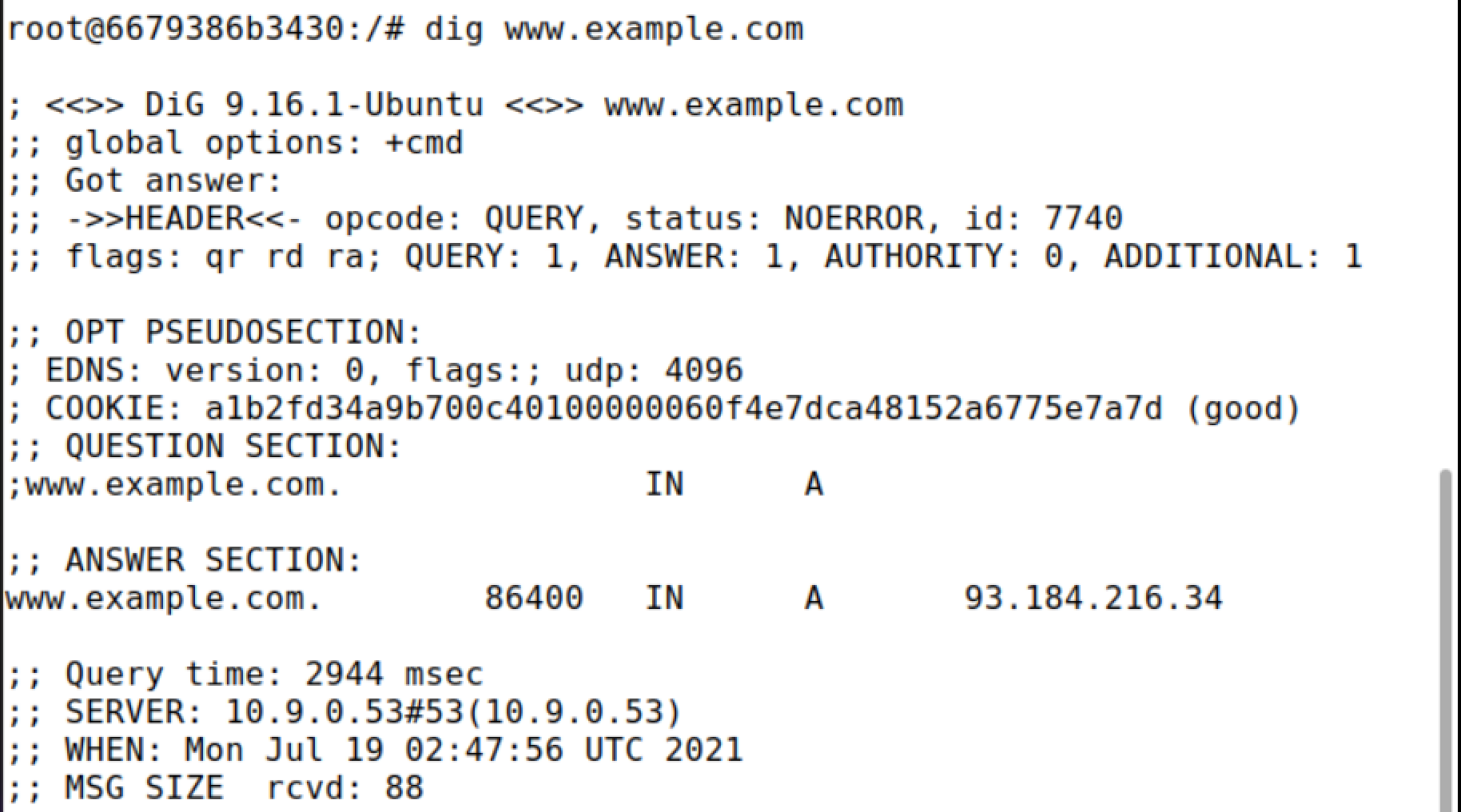
**DNS setup**

（1）获得ns.attacker32.com的IP地址：从结果知，从local dns得到了ns.attacker32.com的IP地址为10.9.0.153。

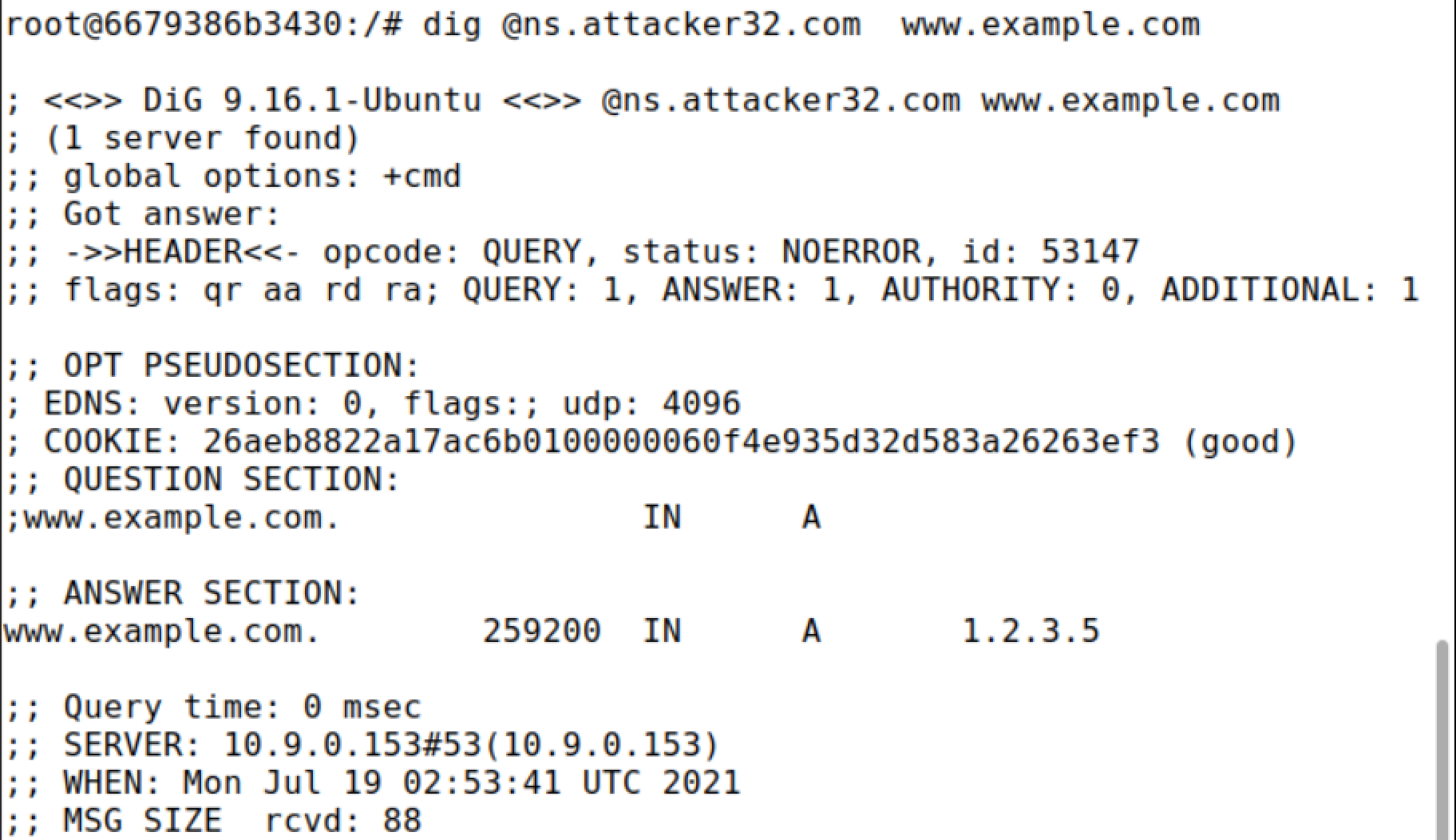


（2）获得www.example.com的IP地址：

dig www.example.com向local dns server询问IP地址，local dns server将从www.example.com的域名服务器获知其IP地址为93.184.216.34。



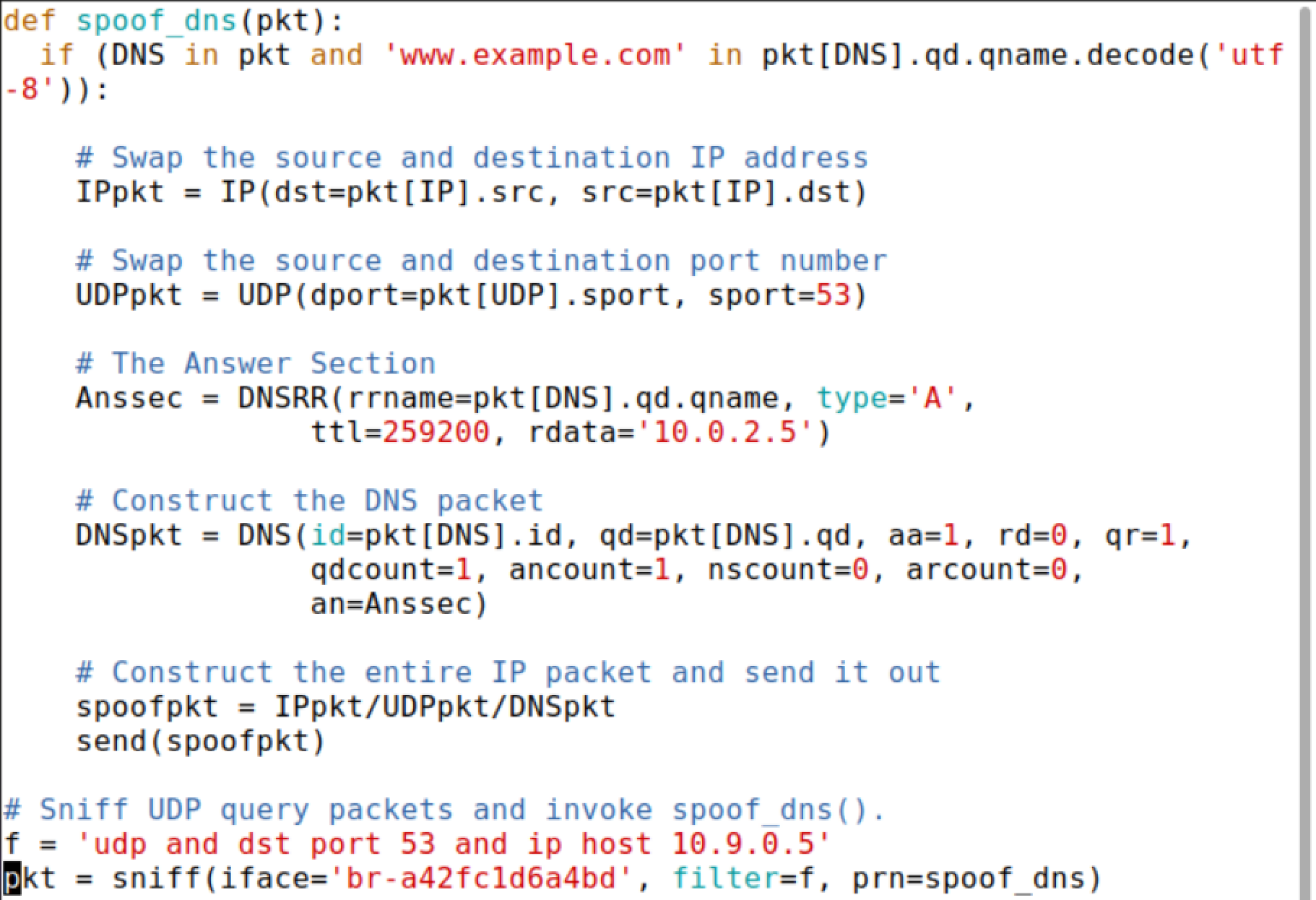
dig @ns.attacker32.com www.example.com向attacker dns server询问IP地址，将直接返回attacker dns server预先设定好的IP地址1.2.3.5。



**Task 1**

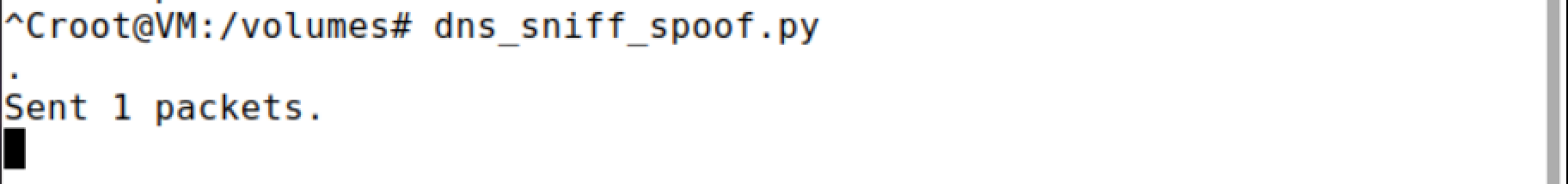
（1）实验内容：实现向用户主机伪造DNS响应报文的DNS攻击，即当用户主机向本地域名服务器请求域名的IP地址时，攻击者抢先向用户主机回复所请求域名IP地址的DNS响应。

（2）代码：修改样例代码中的.net为.com，修改监听的网卡，增加filter条件只抓用户主机的DNS报文。

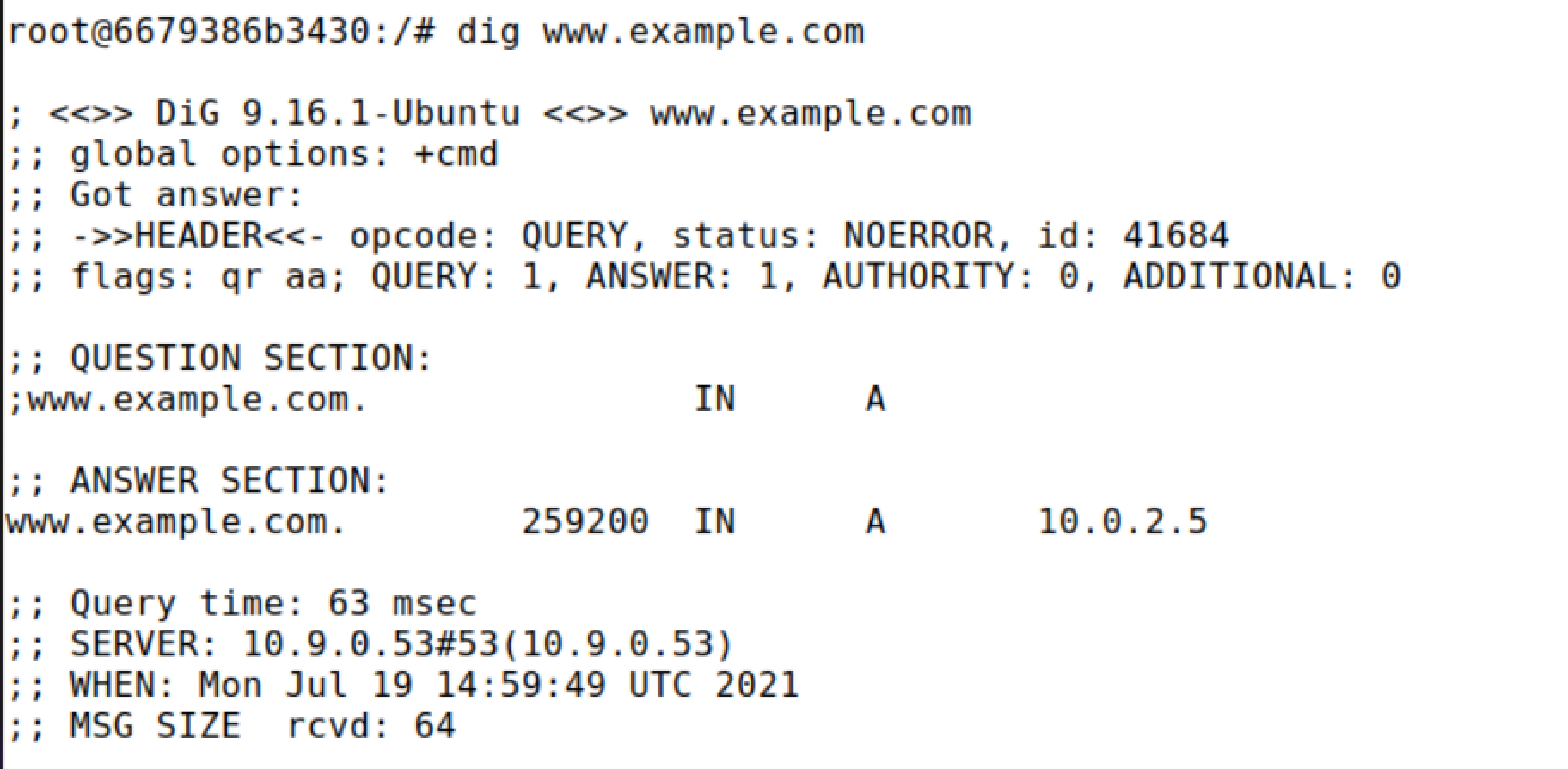


dns\_sniff\_spoof.py

（3）程序执行情况：先执行程序（确保local DNS server cache为空），然后从用户主机查询www.example.com，获得了伪造的DNS响应，证明DNS攻击成功。



dns\_sniff\_spoof.py执行情况

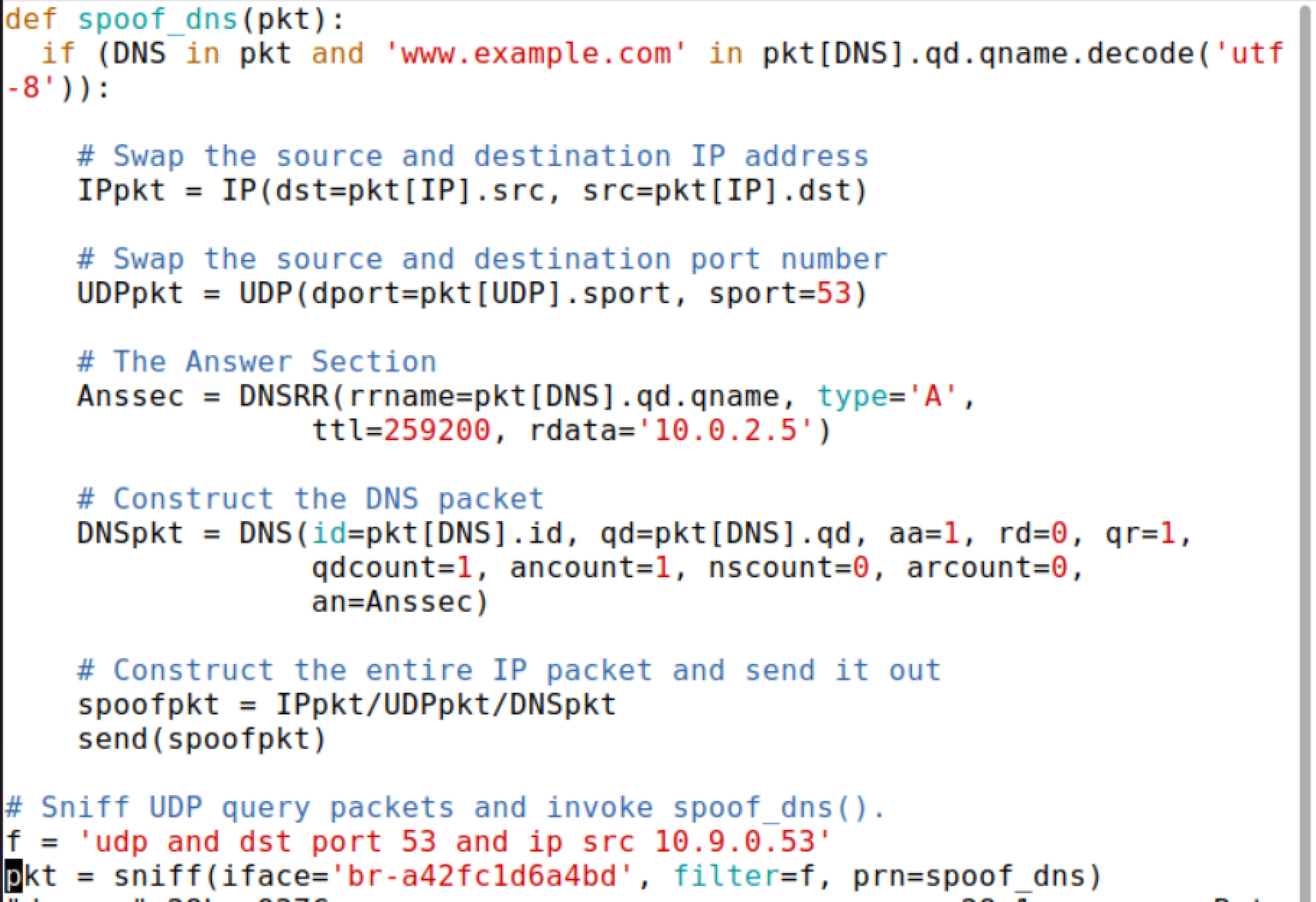


DNS攻击成功，获得伪造的DNS响应

**Task 2**

（1）实验内容：实现向本地域名服务器伪造DNS响应报文的DNS攻击，即当用户主机向本地域名服务器请求域名的IP地址时，本地域名服务器会向外询问，攻击者抢先向本地域名服务器回复所询问额的域名IP地址的DNS响应，以污染本地域名服务器的缓存，由本地域名服务器再将错误的DNS响应返回给用户主机。

（2）代码：修改样例代码中的.net为.com，修改监听的网卡，增加filter条件只抓本地域名服务器的DNS报文。

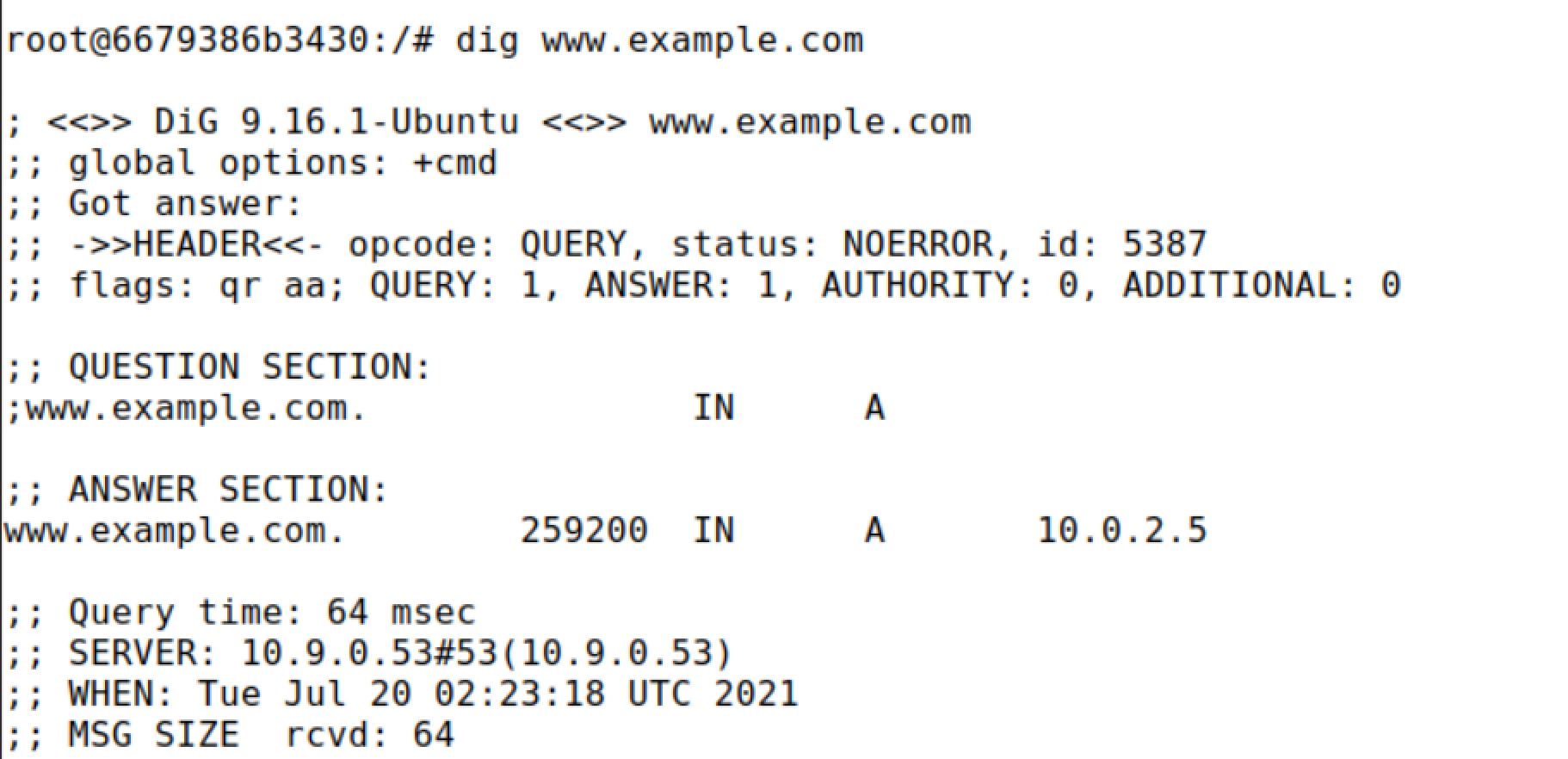


dns\_sniff\_spoof.py

（3）程序执行情况：先清空local DNS server cache，再执行程序，然后从用户主机查询www.example.com，获得了伪造的DNS响应；查看local DNS server cache中包含有伪造的DNS项，证明DNS攻击成功。



dns\_sniff\_spoof.py执行情况

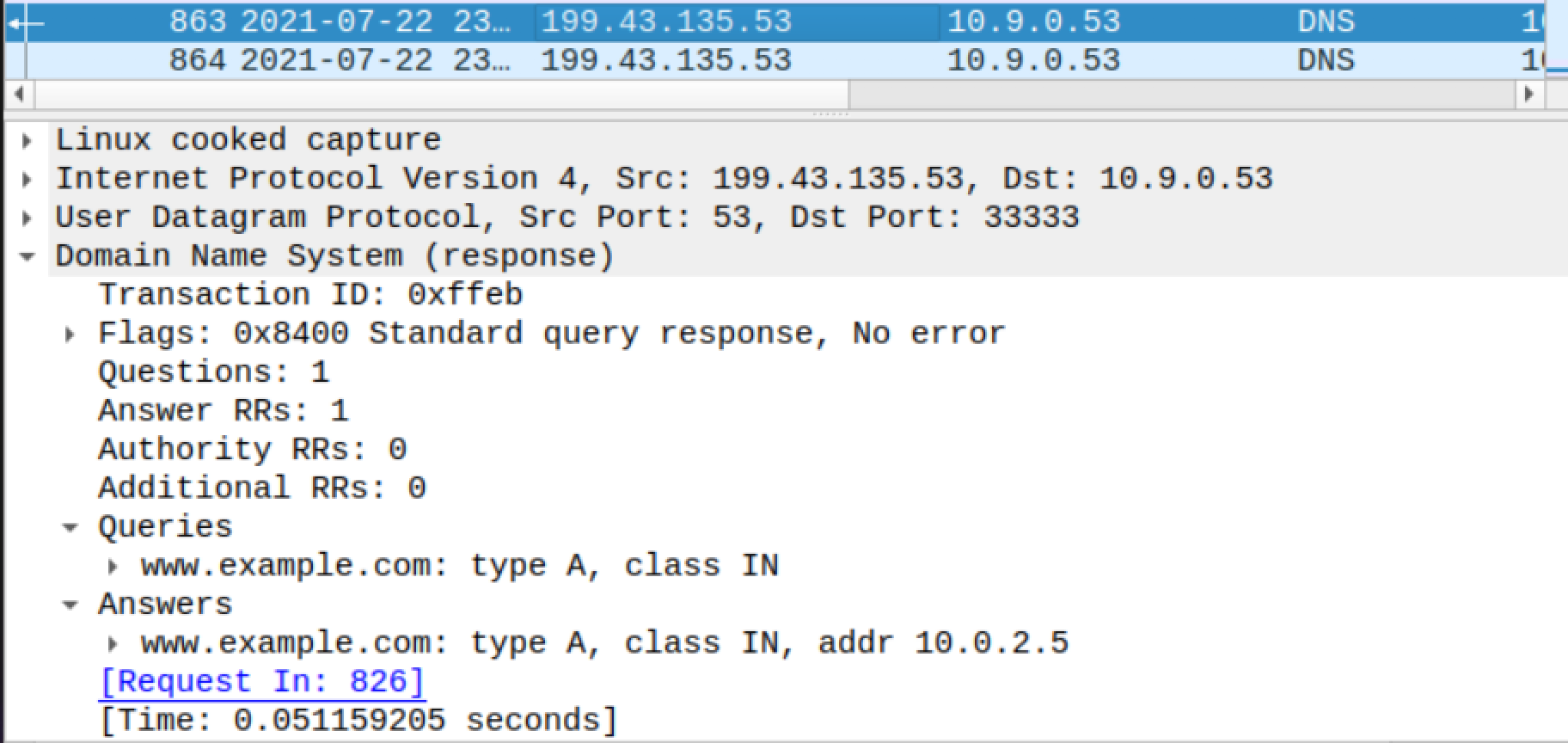


DNS攻击成功，获得伪造的DNS响应

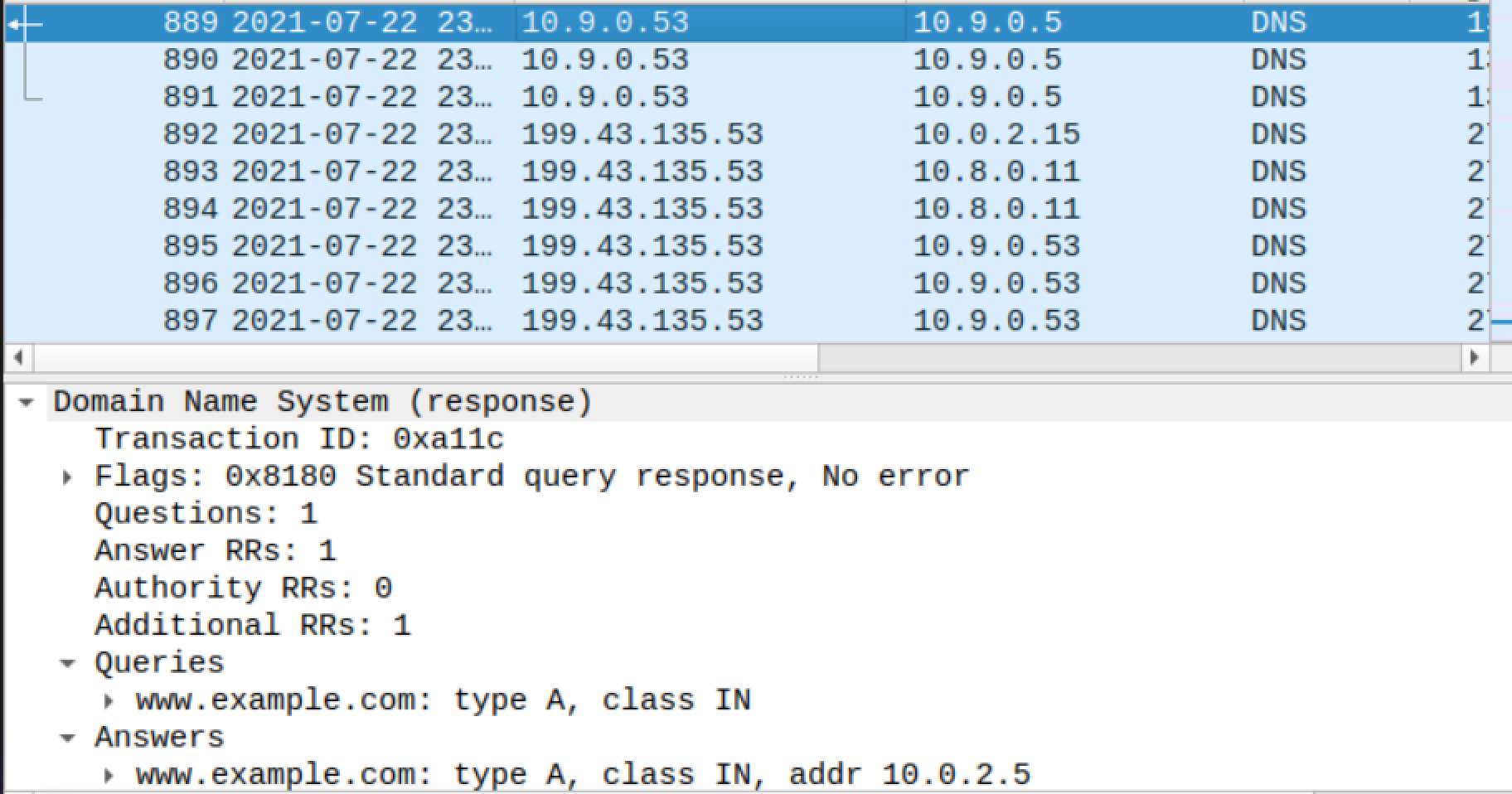


本地域名服务器中的伪造DNS项

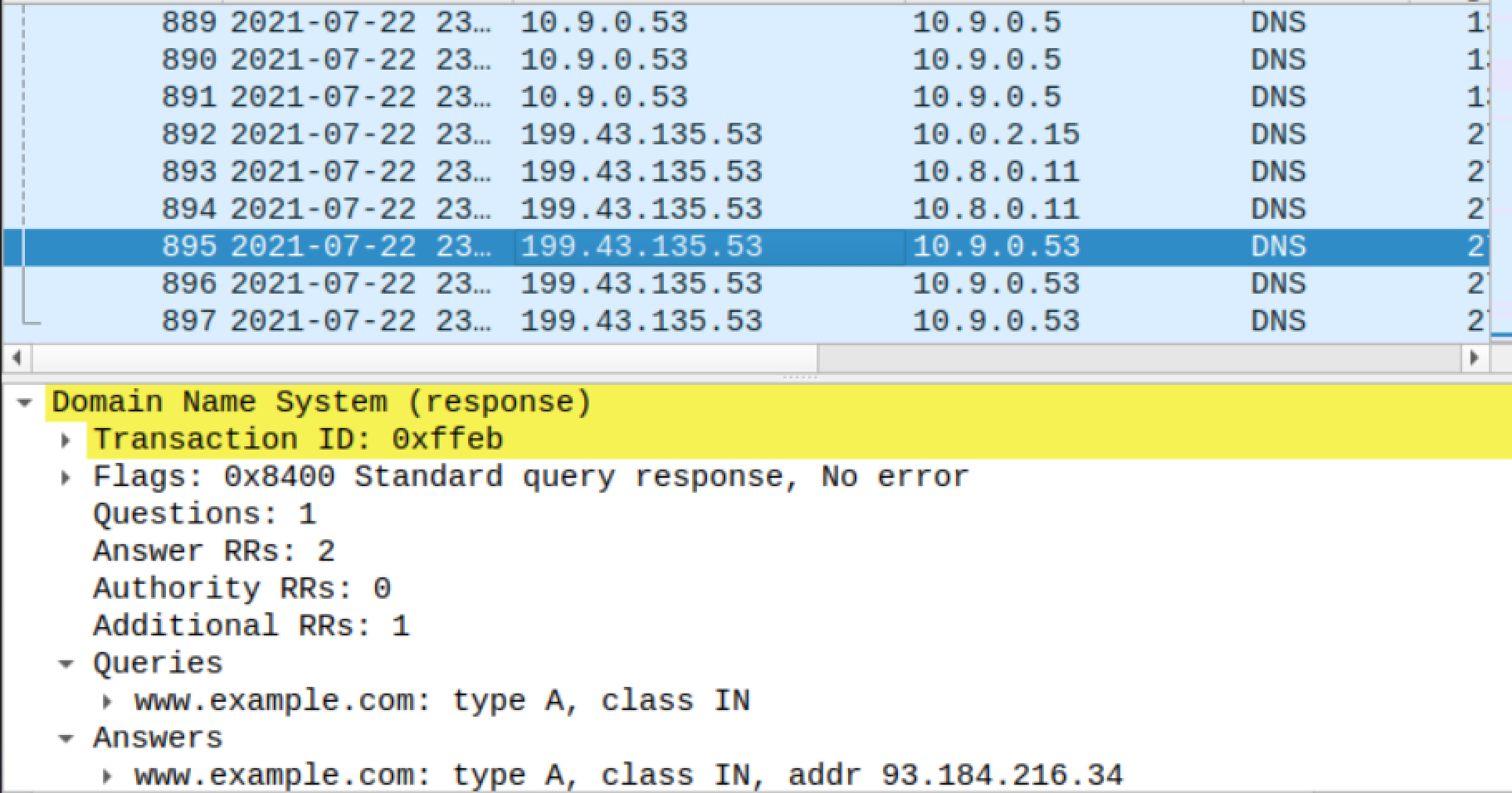
通过Wireshark查看DNS解析和DNS攻击的过程，发现伪造的DNS响应报文抢先发给本地域名服务器，本地域名服务器将其回复给用户主机。之后本地域名服务器才从外得到了正确的DNS响应报文，该正确的DNS响应报文被认为是重复的响应报文，本地域名服务器将不予理会。



本地域名服务器获得伪造的DNS响应



本地域名服务器向用户主机返回伪造的DNS响应



本地域名服务器在伪造报文后获得正确的DNS响应

**Task 3**

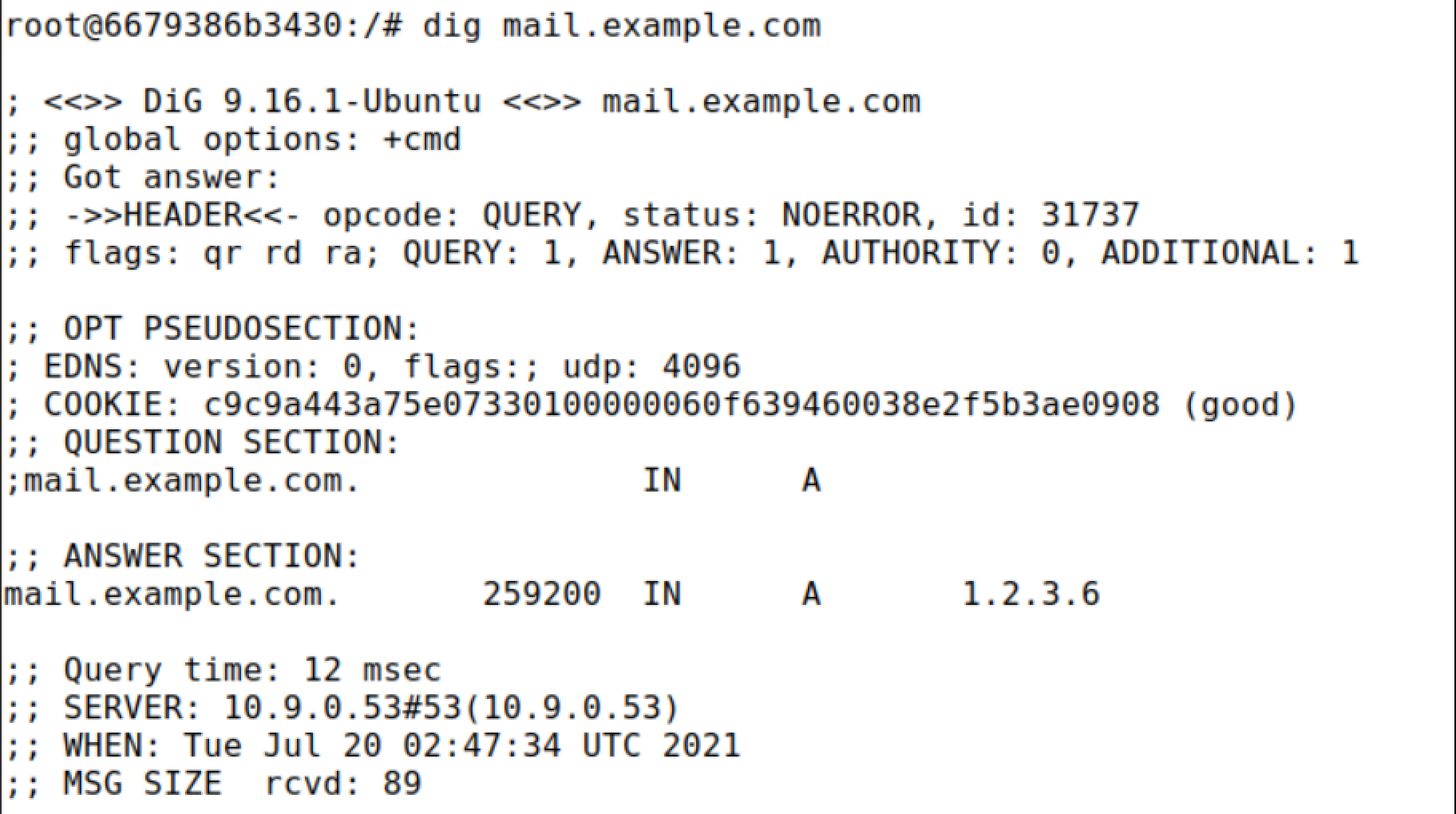
（1）实验内容：采用面向本地域名服务器的攻击方法，在伪造的DNS报文中添加AUTHORITY SECTION，以使在example.com域内的地址都由交由attacker router解析。

（2）代码：在Task 2中的代码增加AUTHORITY SECTION段。



dns.py

（3）程序执行情况：先清空local DNS server cache，再执行程序，然后从用户主机查询www.example.com，获得了伪造的DNS响应，以完成DNS攻击。再在从用户主机查询mail.example.com，得到了attacker中设置好的地址1.2.3.6。

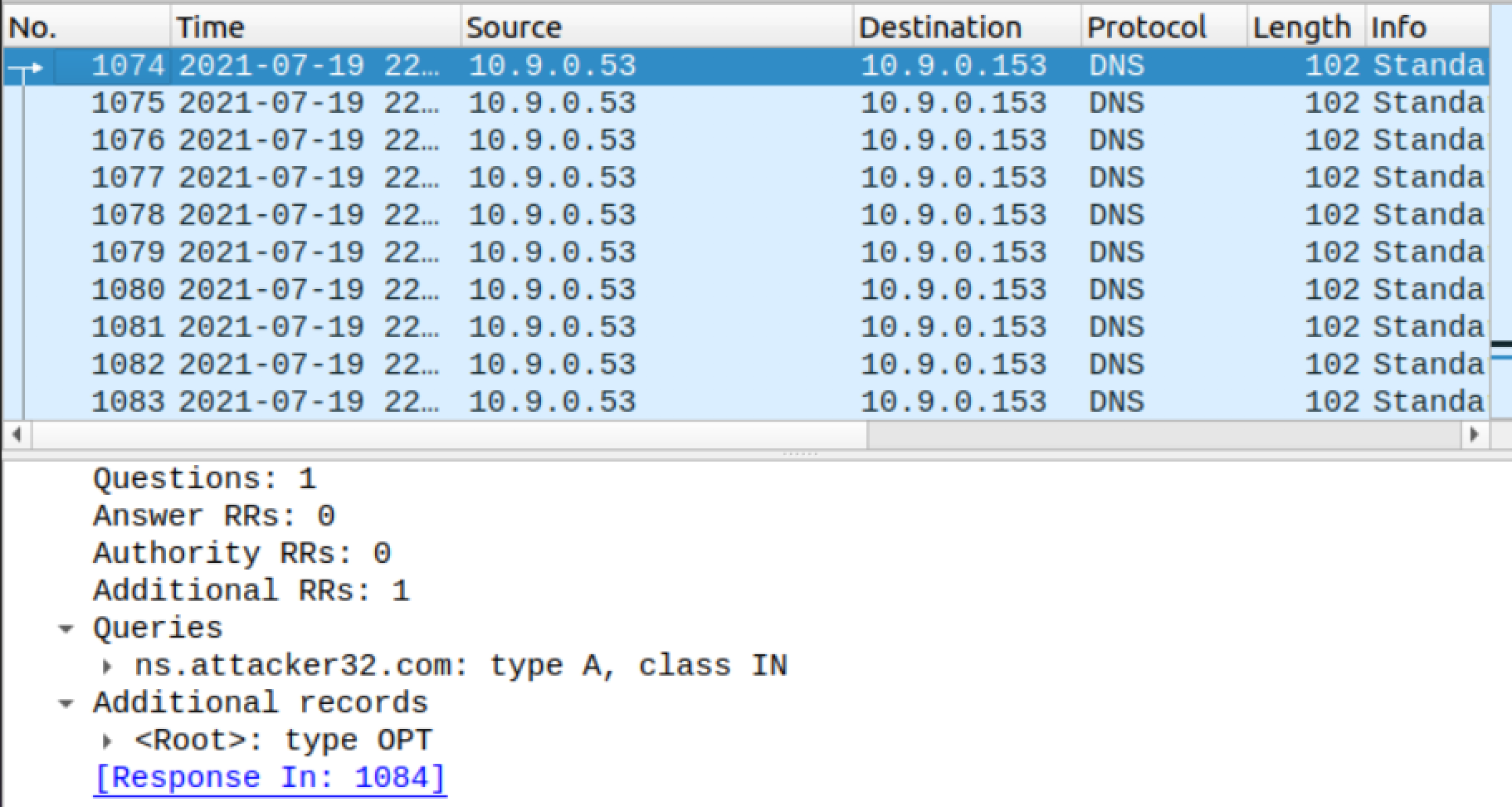


查询example.com域内地址，直接重定向



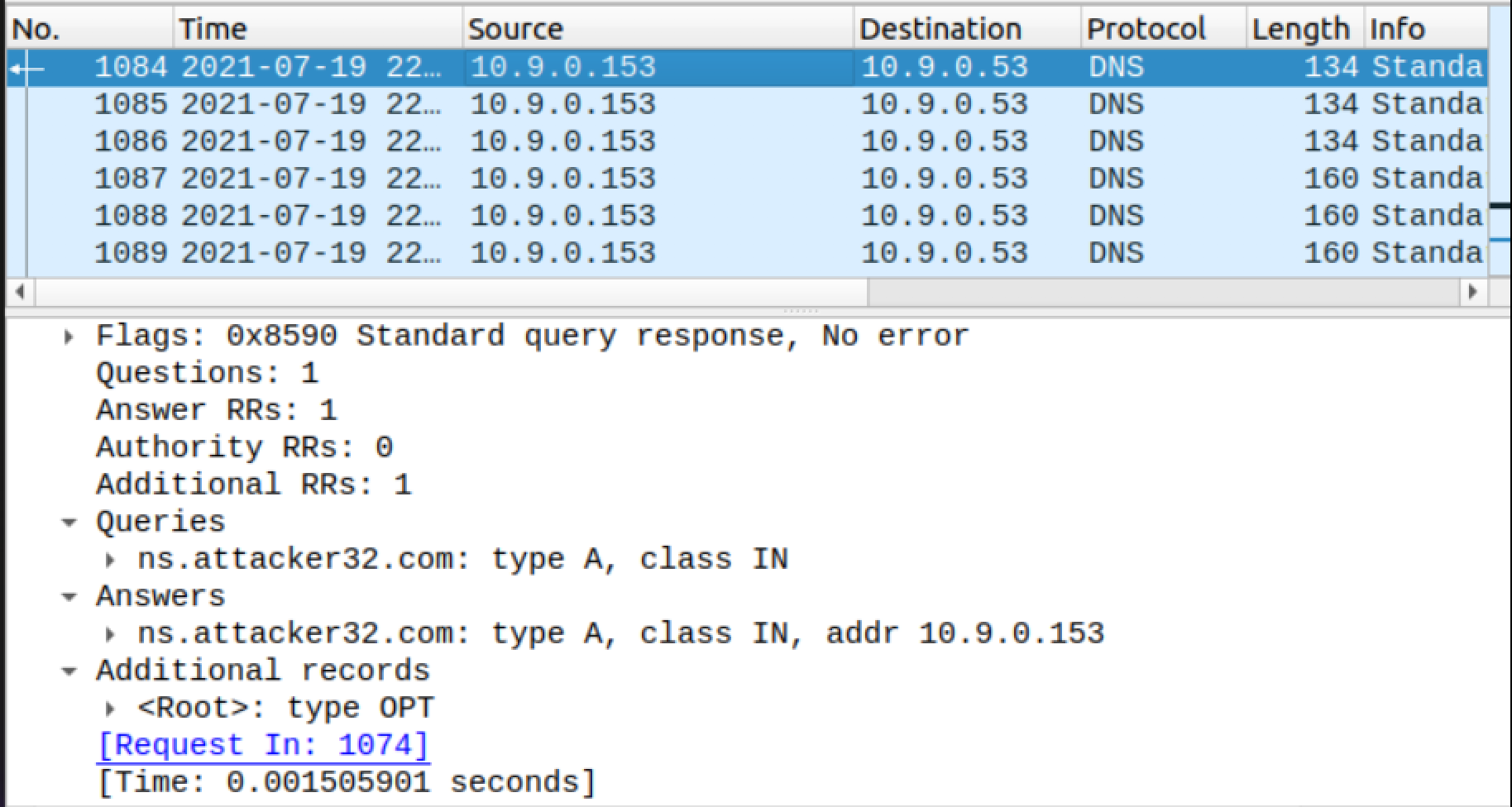
ns记录出现于local DNS server cache中

通过Wireshark发现，用户主机向本地域名服务器发出查询请求；本地域名服务器根据cache中的NS项得知应询问ns.attacker32.com，根据配置，attacker.com域内的地址应询问10.9.0.153，本地域名服务器向10.9.0.153发出请求；



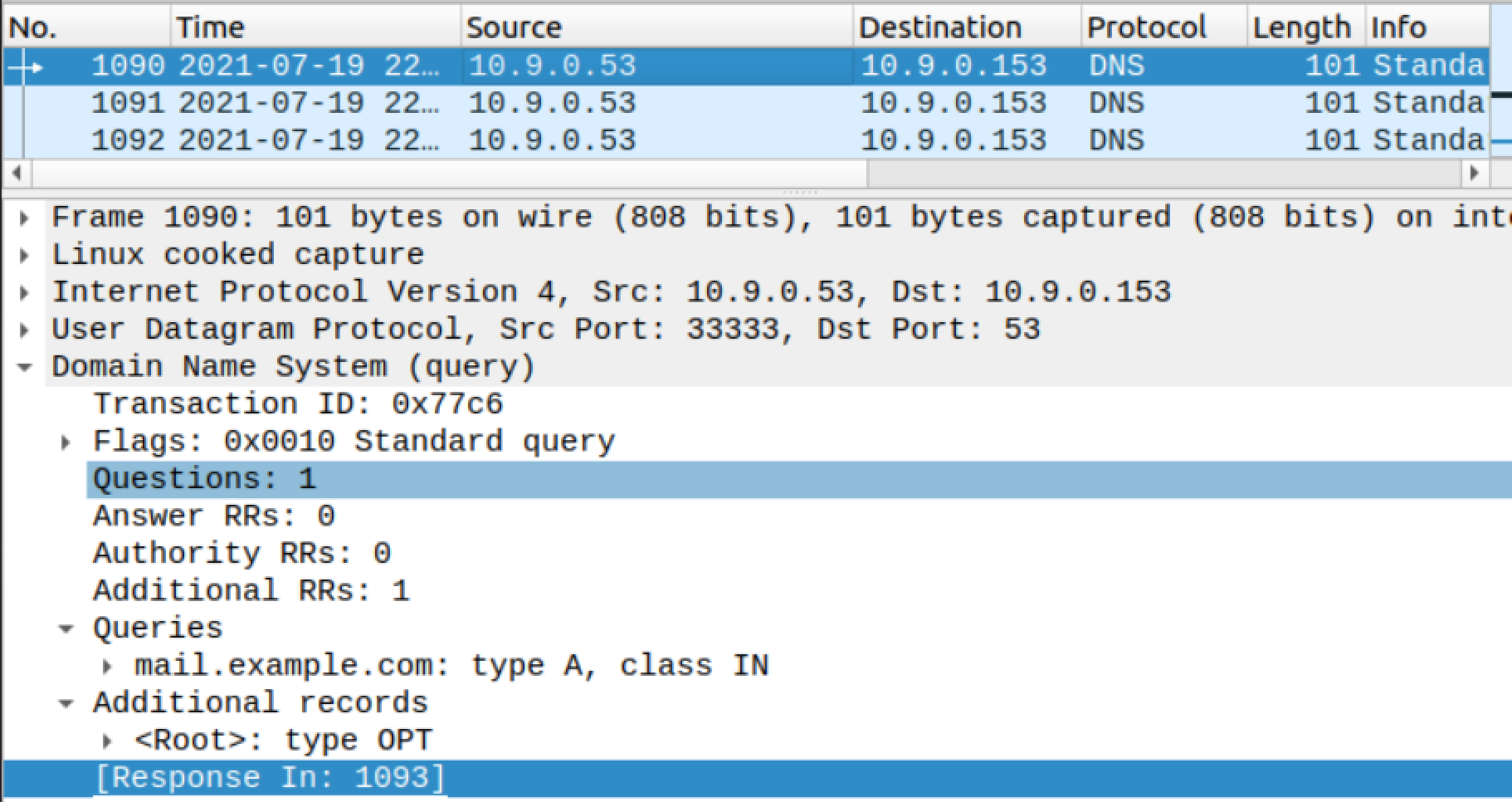
本地域名服务器向10.9.0.153询问ns.attacker32.com

10.9.0.153返回请求，本地域名服务器得知ns.attacker32.com的地址仍为10.9.0.153；

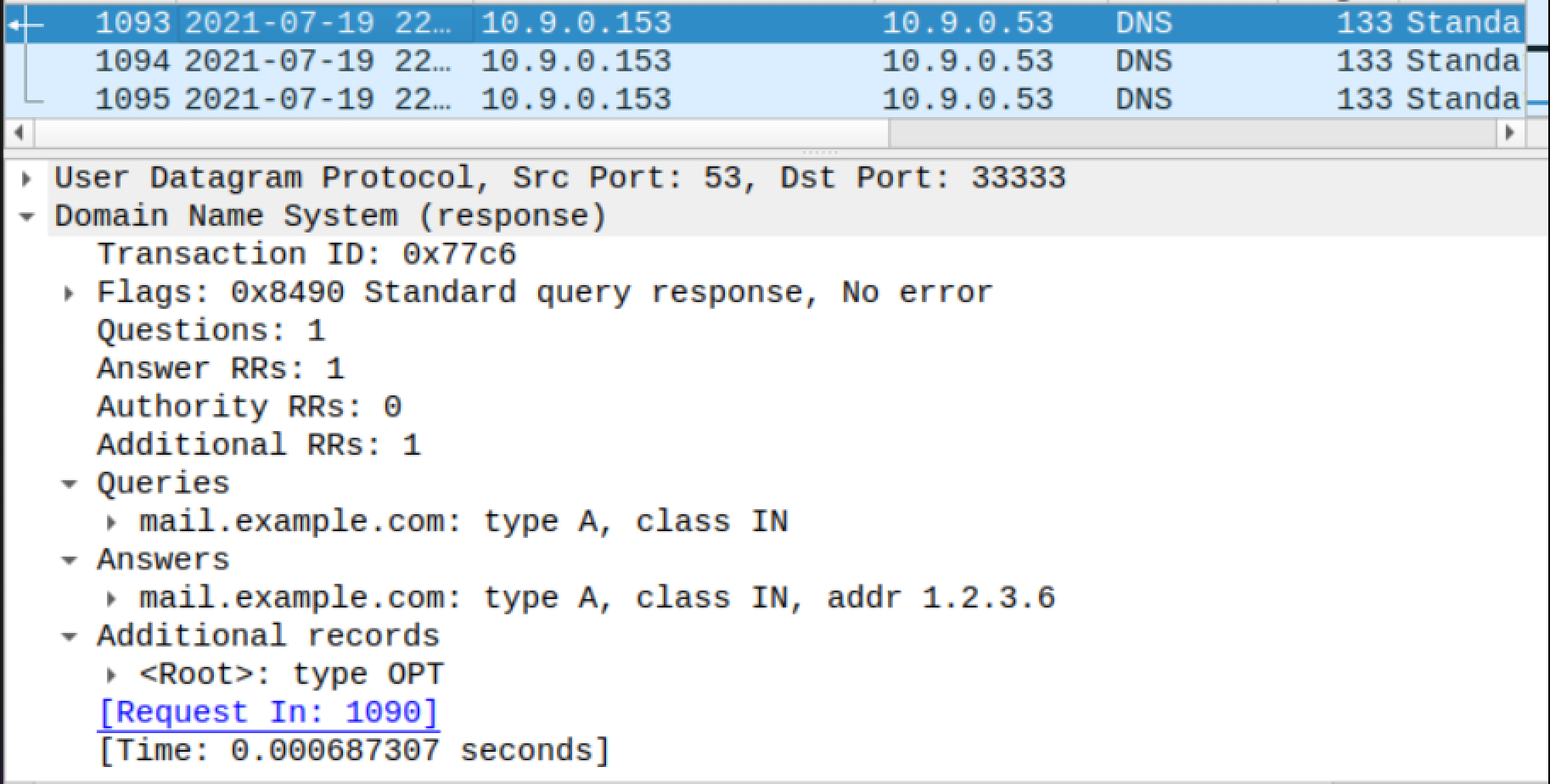


10.9.0.153返回ns.attacker32.com地址

再向10.9.0.153询问mail.example.com地址，最后10.9.0.153返回地址。



本地域名服务器向ns.attacker32.com询问mail.example.com

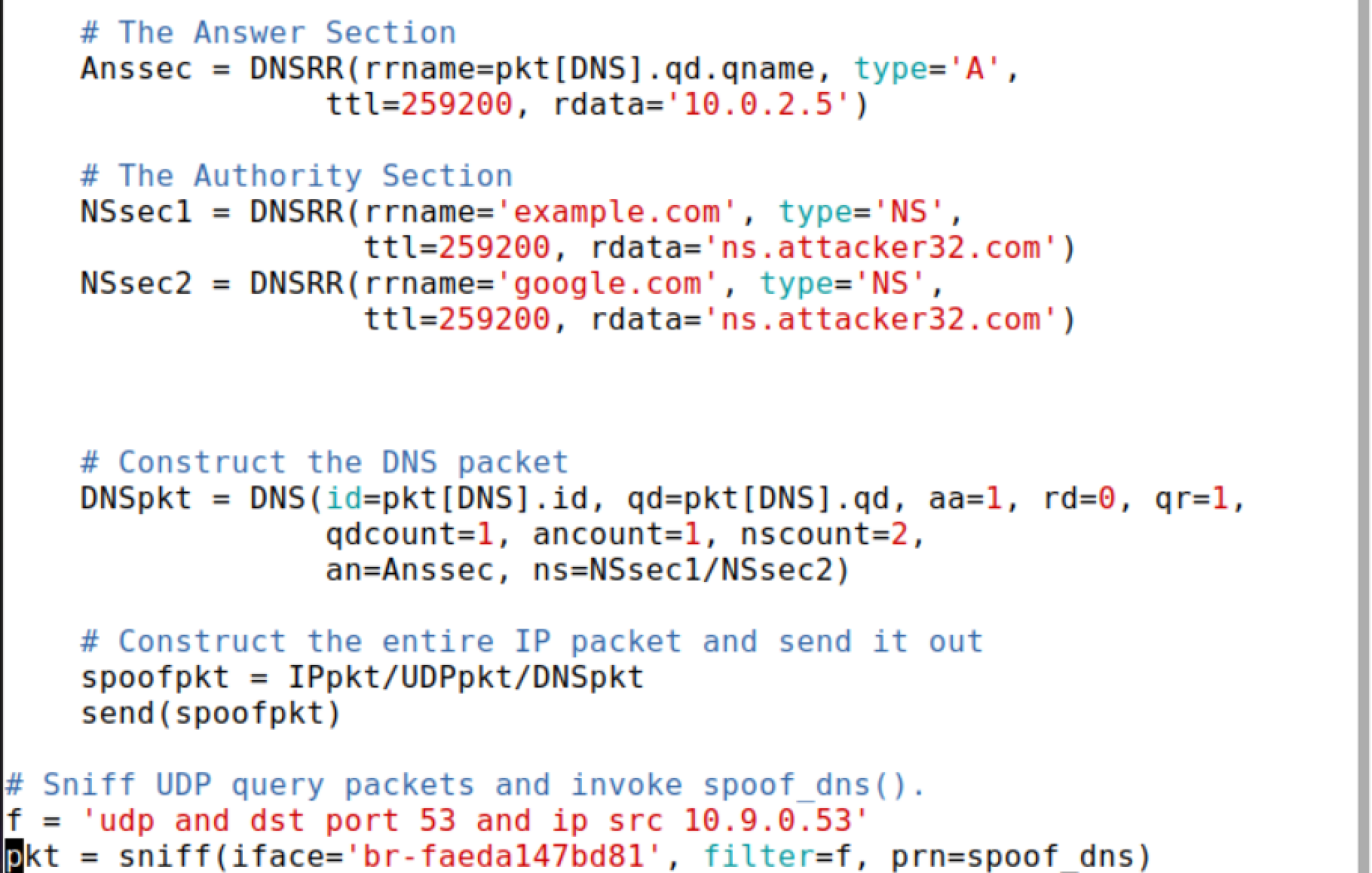


ns.attacker32.com返回mail.example.com地址

**Task 4**

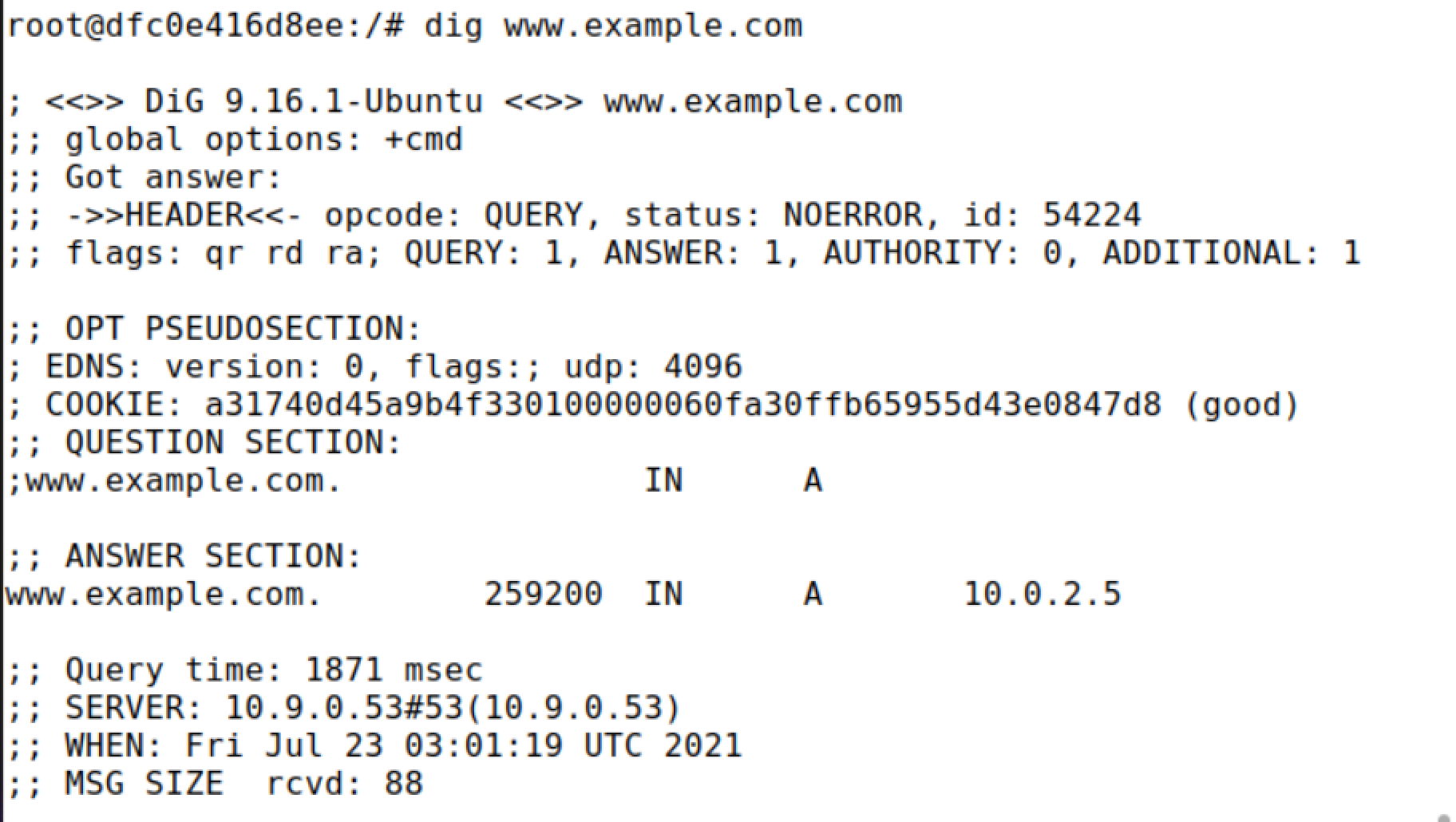
（1）实验内容：采用面向本地域名服务器的攻击方法，在伪造的DNS报文中添加非example.com域的AUTHORITY SECTION，探究其是否会被local DNS server接收。

（2）代码：在Task 3的代码中添加关于google.com的NS项。



dns.py

（3）程序执行情况：先清空local DNS server cache，再执行程序，然后从用户主机查询www.example.com，由于采用面向本地域名服务器的攻击方法，用户主机得到的是从本地域名服务发来的只含有ANSWER SECTION的报文。查看local DNS server cache中并没有google.com的NS项，即它并没有被cache记录。这是因为查询的域名并不是google.com下所属的域名，所以该NS记录不会被保存。



用户主机得到伪造的DNS响应

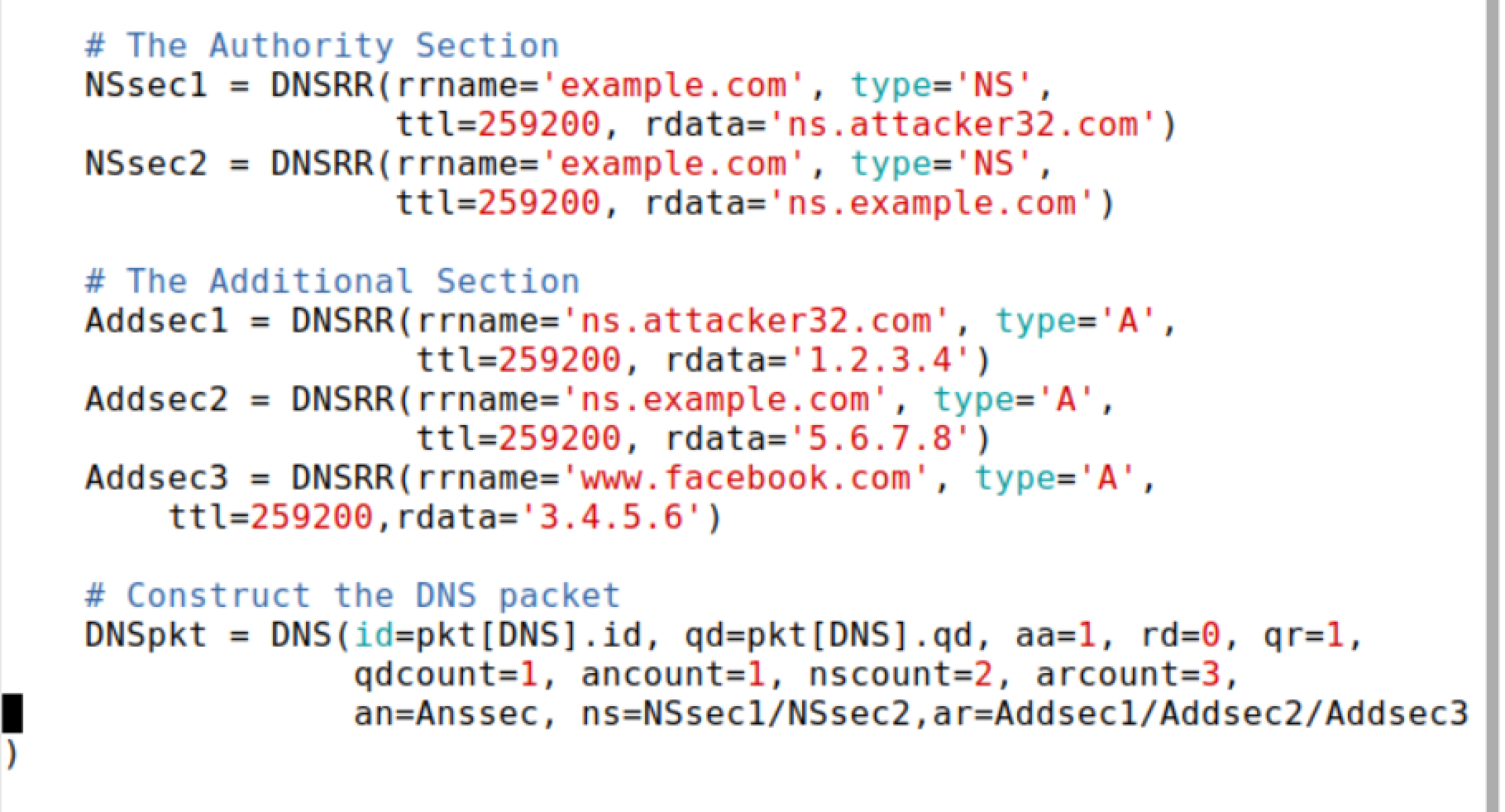


local DNS server cache中并无新增的NS项

**Task 5**

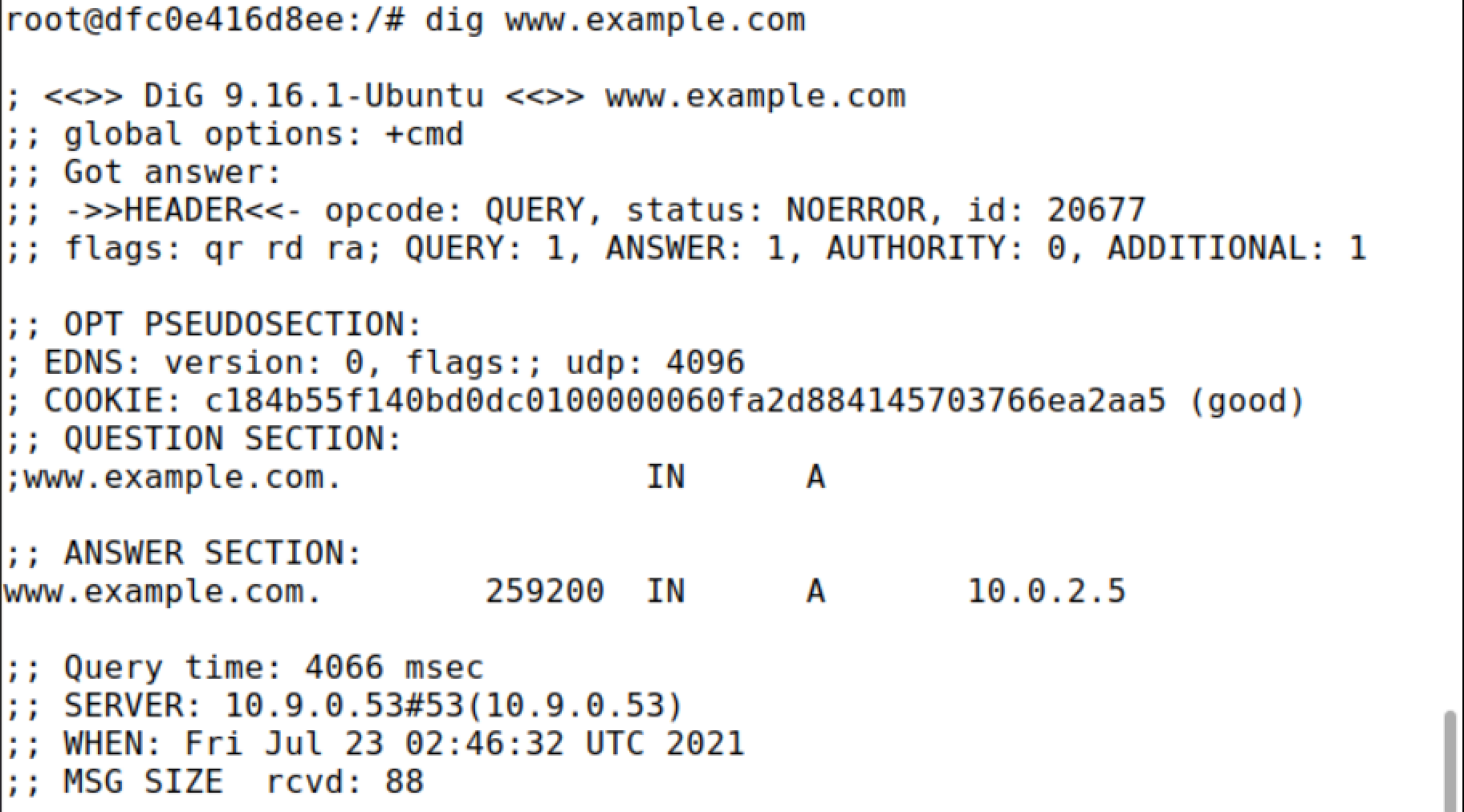
（1）实验内容：采用面向本地域名服务器的攻击方法，在伪造的DNS报文中添加多个AUTHORITY SECTION项与ADDITIONAL SECTION项，探究哪些项会被local DNS server接收。

（2）代码：



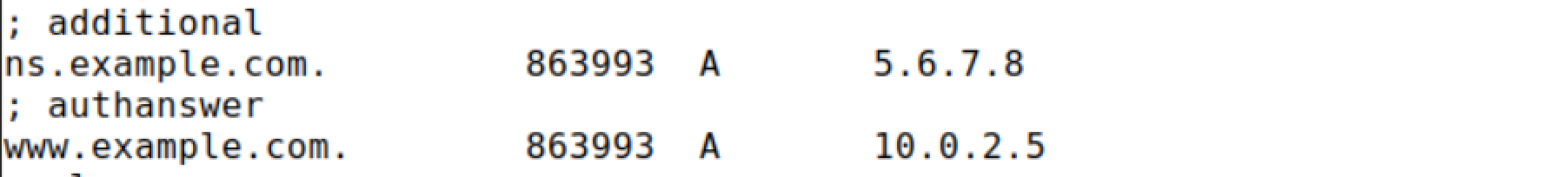
dns.py

（3）程序执行情况：先清空local DNS server cache，再执行程序，然后从用户主机查询www.example.com，由于采用面向本地域名服务器的攻击方法，用户主机得到的是从本地域名服务发来的只含有ANSWER SECTION的报文。

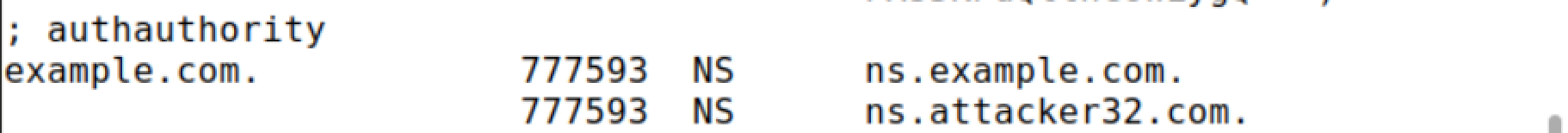


用户主机得到伪造的DNS响应

查看local DNS server cache中并不存在ns.attacker.com的A记录与www.facebook.com的A记录。不存在ADDITTIONAL SECTION中ns.attacker.com的A记录，可能是因为ns.example.com在example.com域中，其A记录可信，而ns.attacker.com不在example.com域中，虽然权威域名服务器可以不在example.com域中，但其附加的A记录不可信，所以不被接收；不存在www.facebook.com的A记录，是因为它不在example.com域内，且不在AUTHORITY SECTION项内，所以不被接收。



local DNS server cache信息



local DNS server cache信息