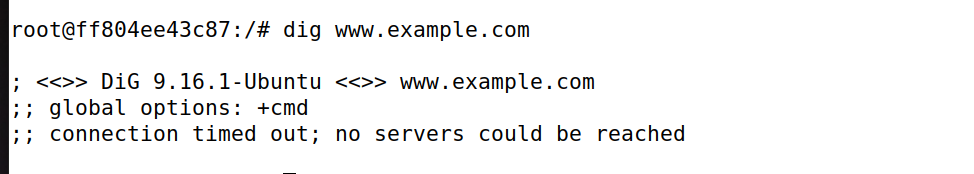
**Local DNS Attack Lab**

**Testing the DNS Setup**

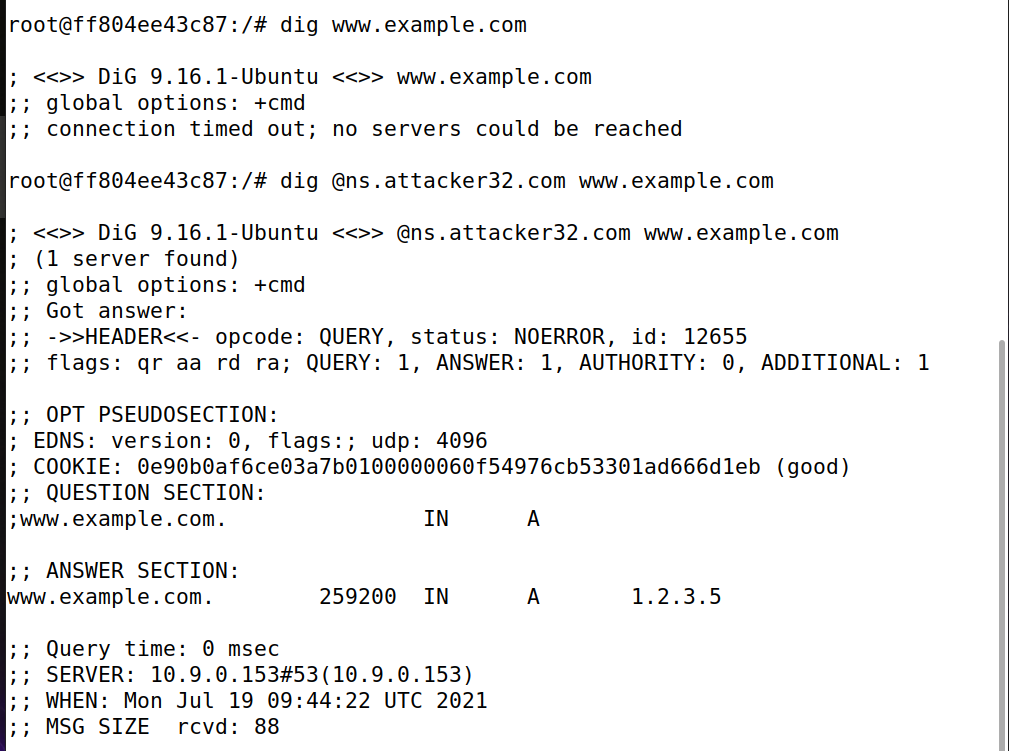
**在受害主机10.9.0.5上运行dig命令，得到ns.attacker32.com的ip地址为10.9.0.153**



**执行命令dig** [**www.example.com**](http://www.example.com)**，发现无法获得想要的ip地址**



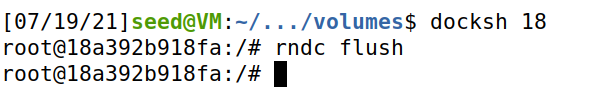
**执行命令@ns.attacker32**



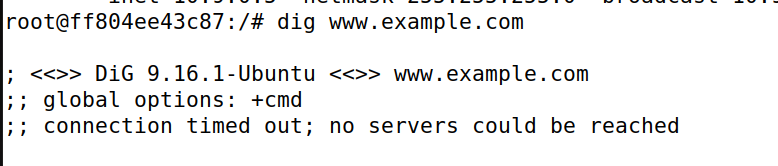
**桥接模式下正常dig无响应**

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

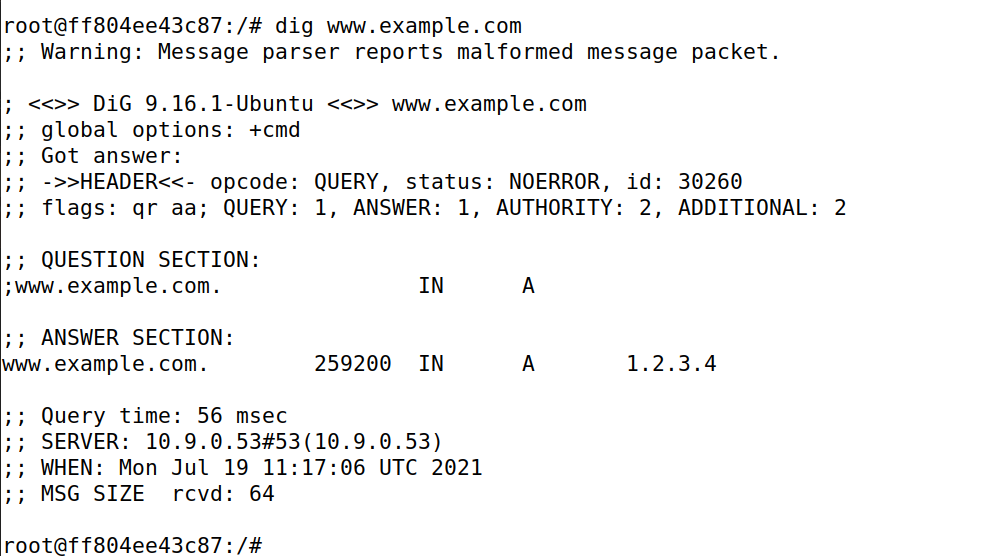
**首先在本地DNS服务器上清除DNS缓存**

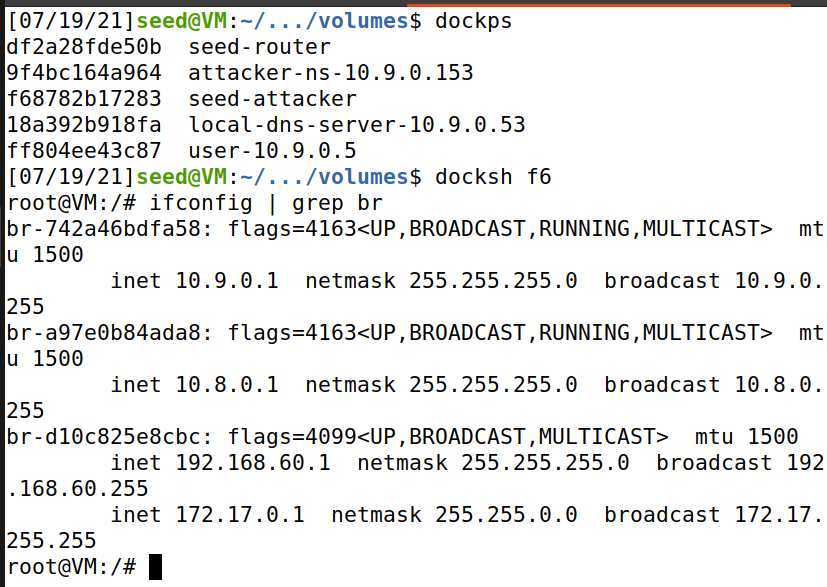


**未执行攻击程序之前，dig www.example.com 会超时无响应**

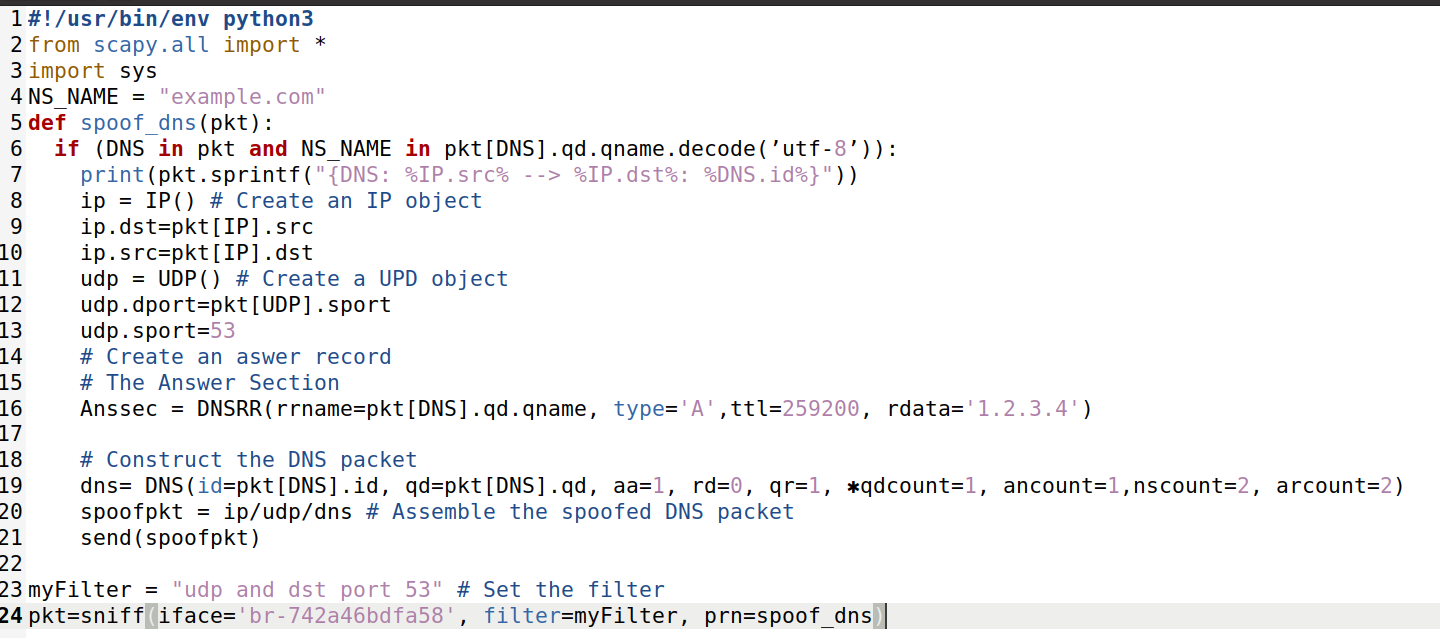


**运行脚本后，结果如下**



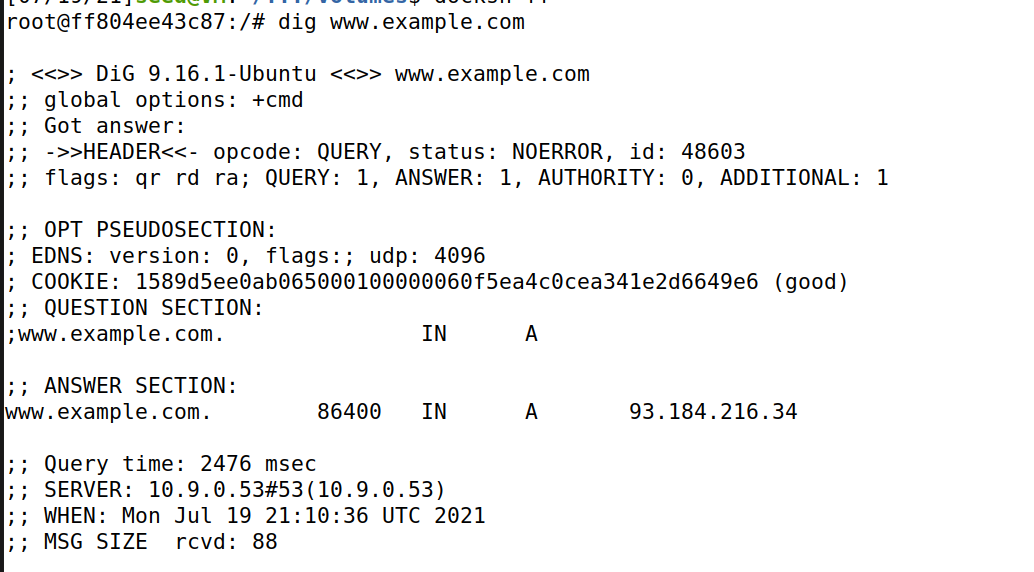


**程序代码如下：**

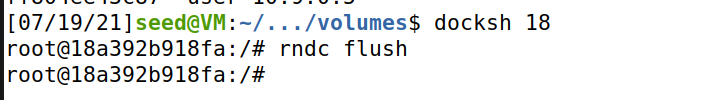


**Task 2: DNS Cache Poisoning Attack – Spoofing Answers**

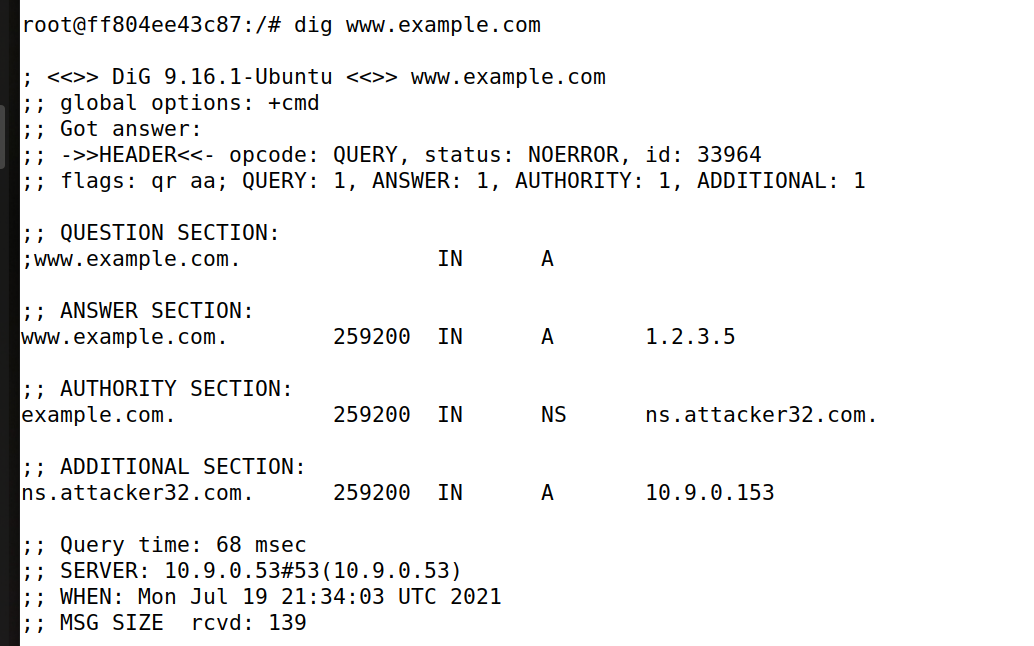
切换成NAT模式，未进行攻击前



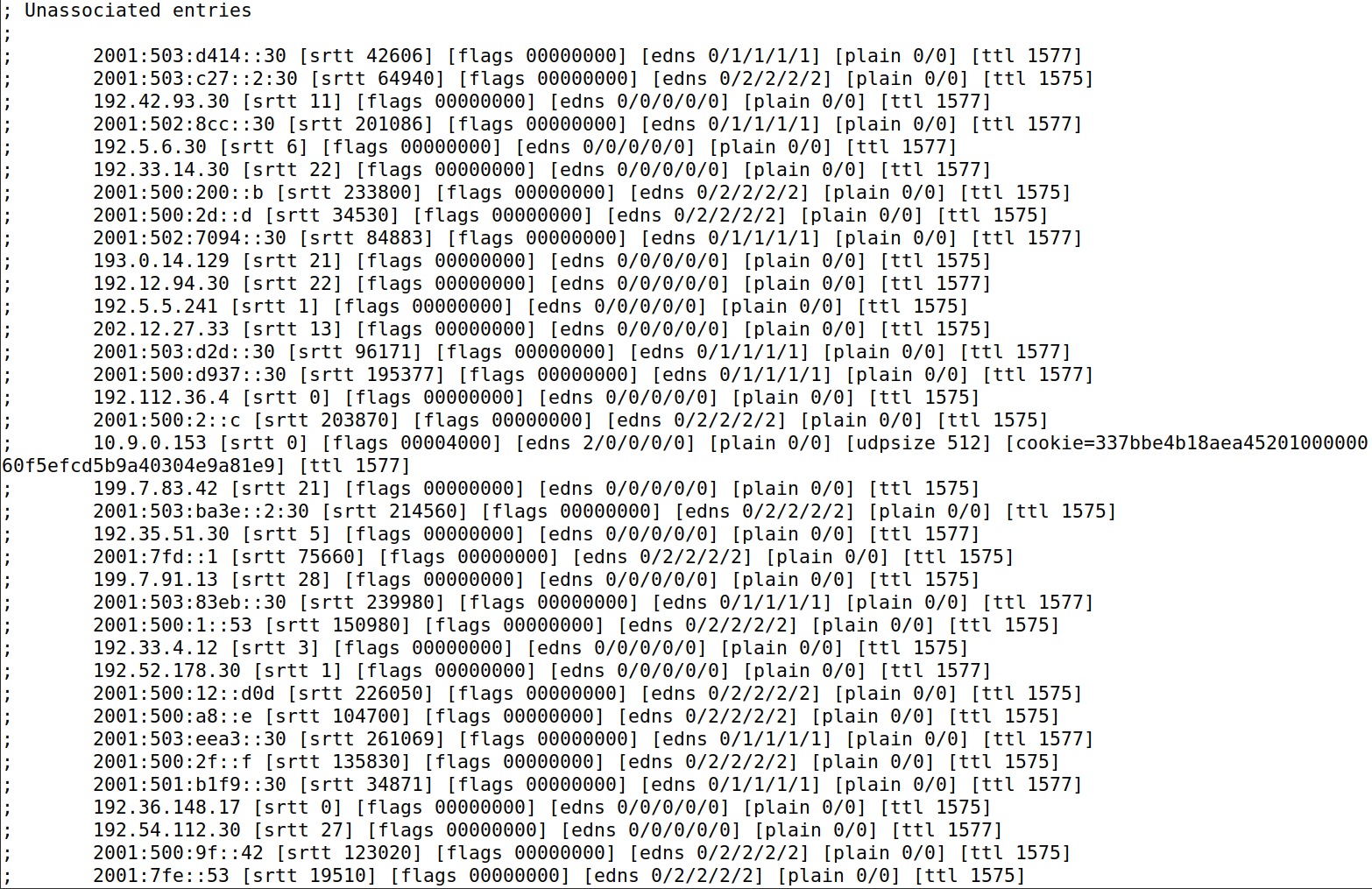
本地DNS中清除缓存

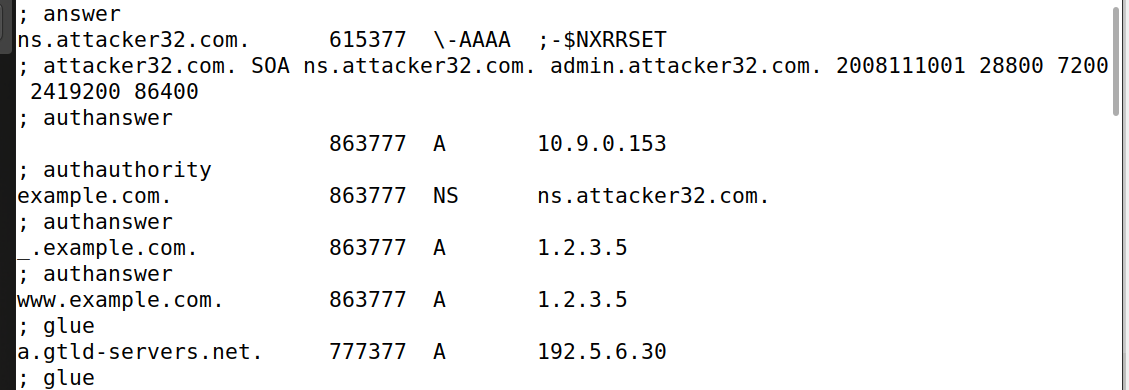


运行攻击程序，结果如下：

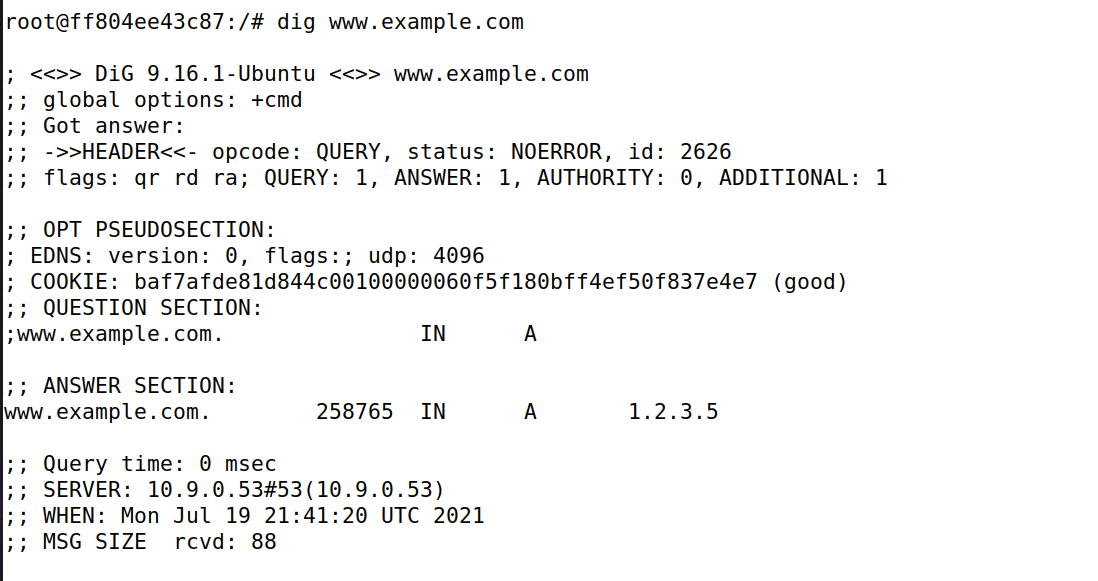


查看DNS缓存

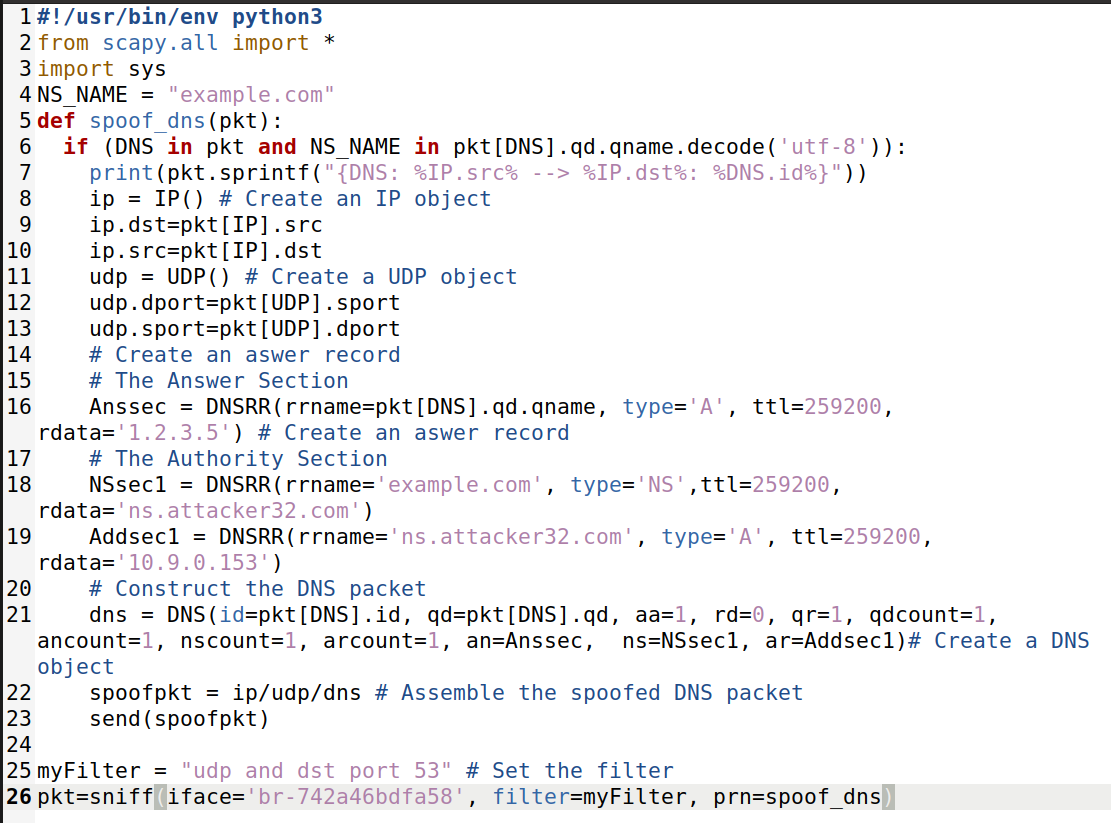




关闭程序测试，结果如下说明已写入缓存中

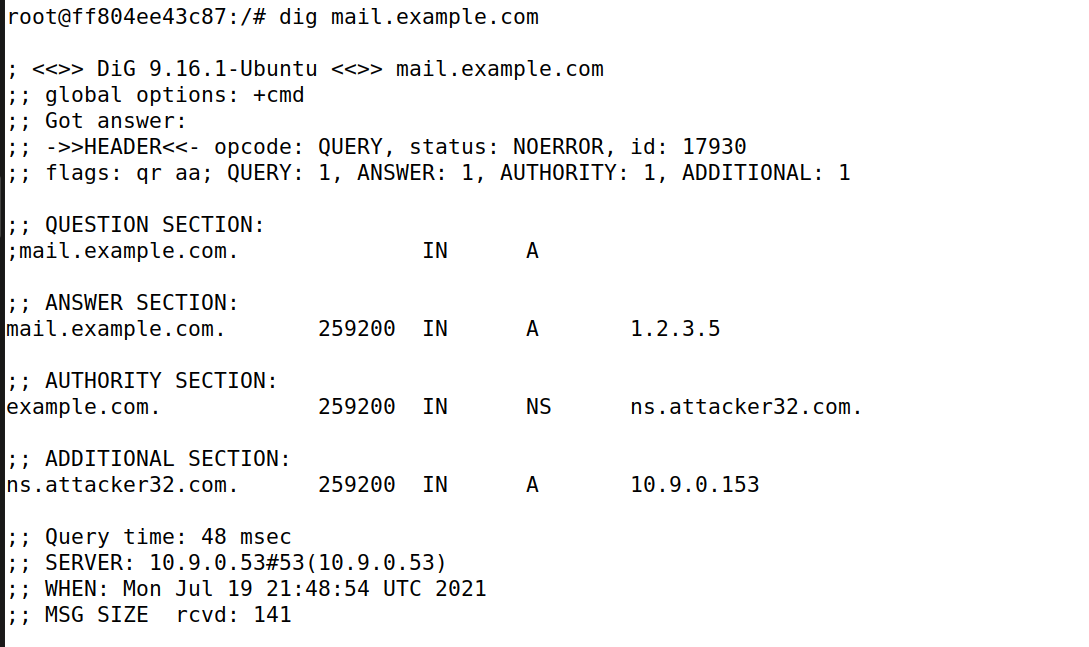


攻击程序代码如下：

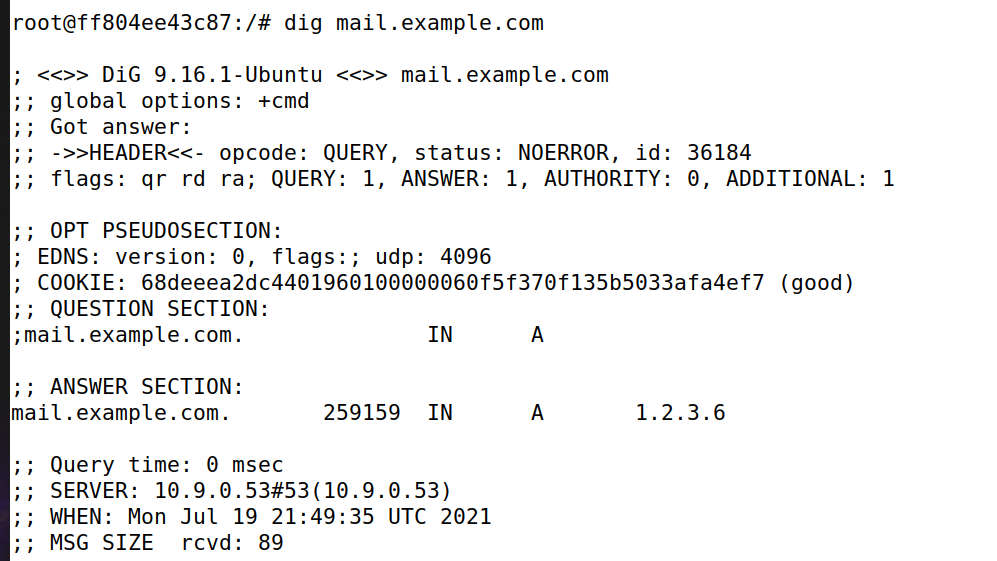


**Task 3: Spoofing NS Records**

清除本地DNS缓存，然后执行程序，结果如下

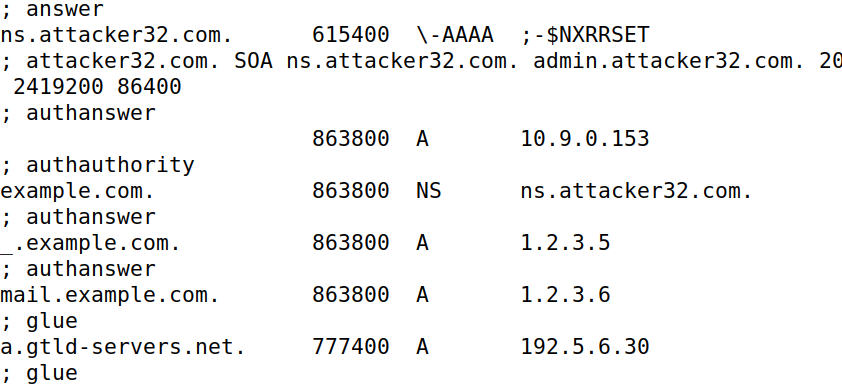


关闭程序测试，结果如下：



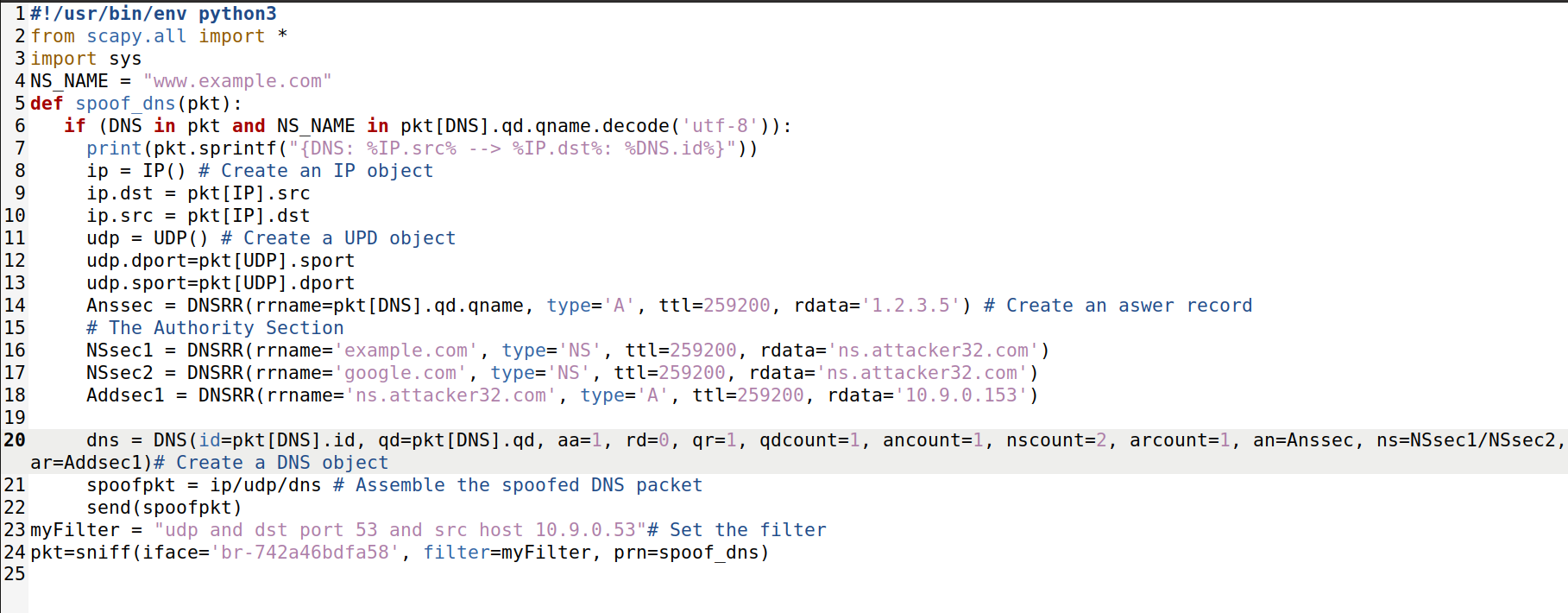
运行程序代码与Task 2一致

查看缓存，发现DNS数据已经写入

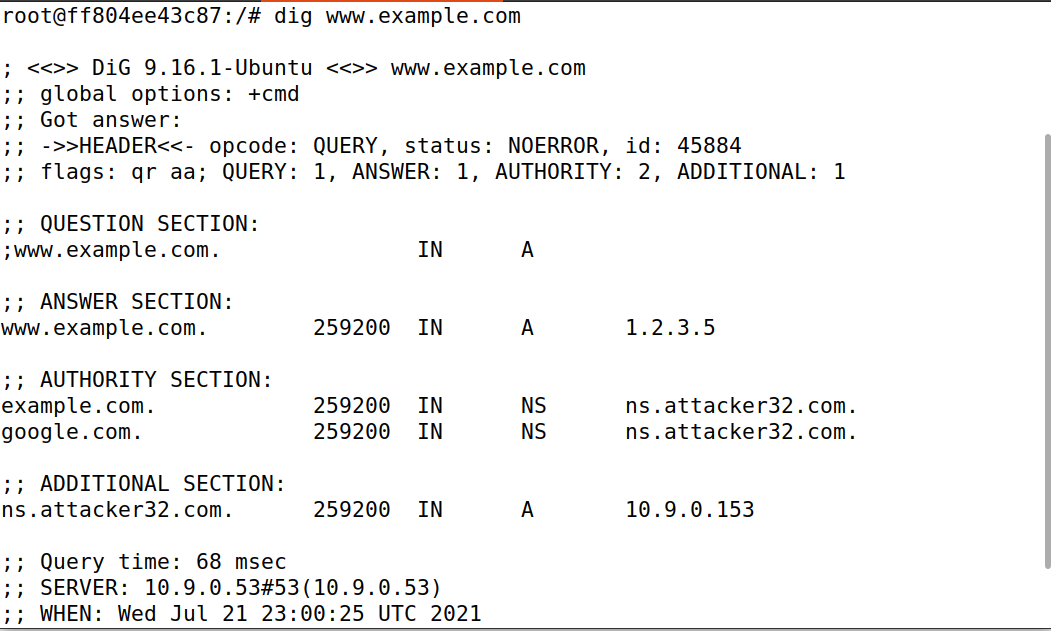


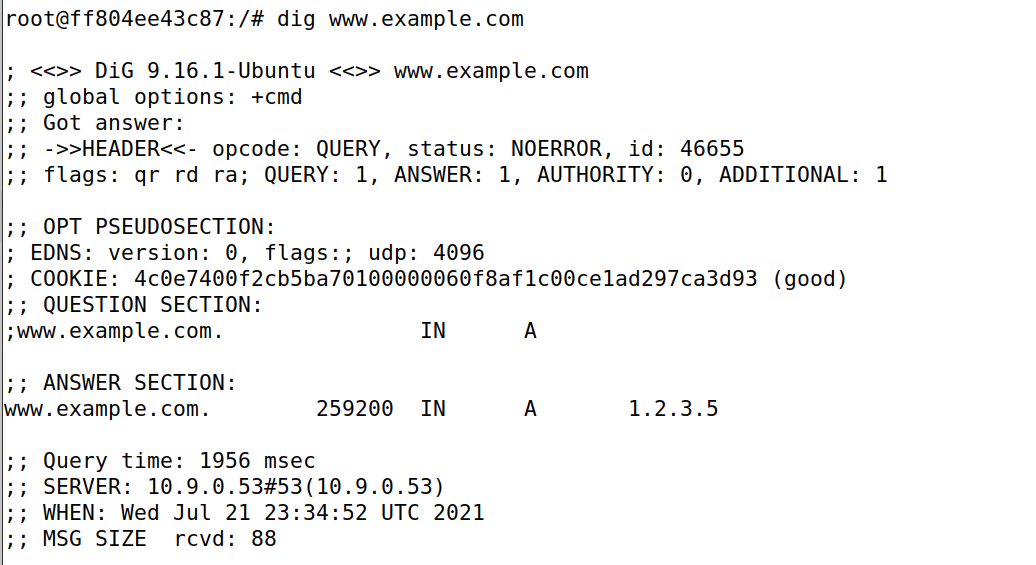
**Task 4: Spoofing NS Records for Another Domain**

清除本地DNS服务器的缓存，然后执行以下攻击程序，代码如下：

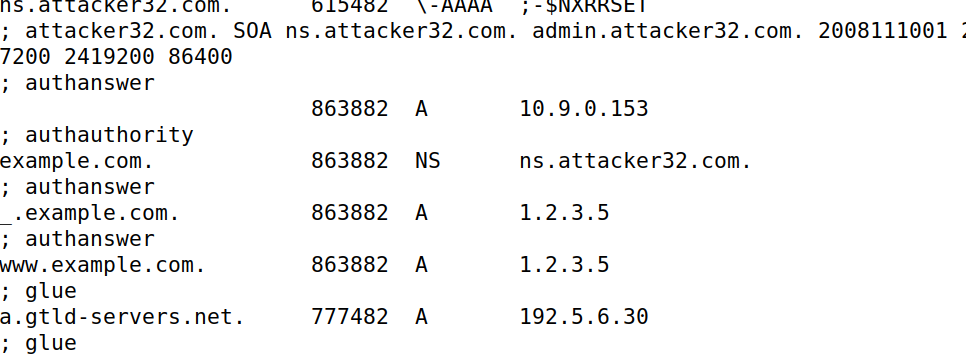


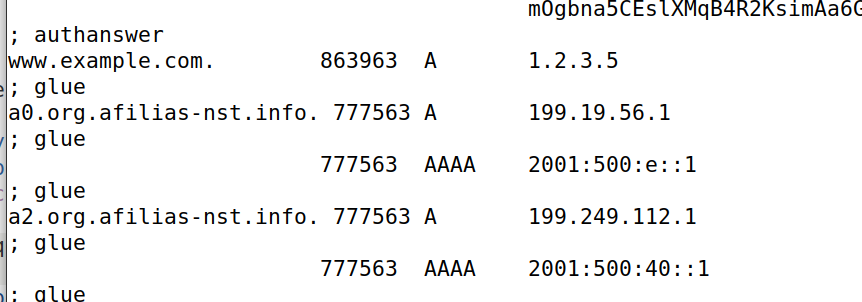
实验结果如下：



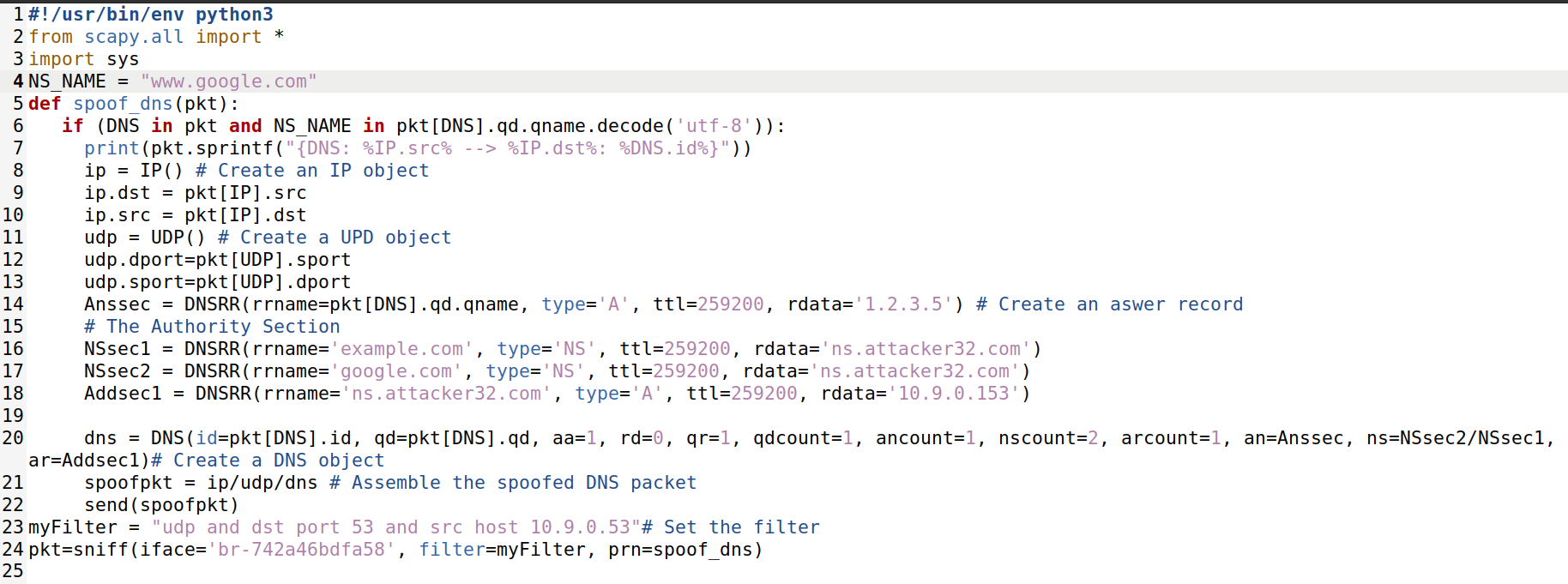


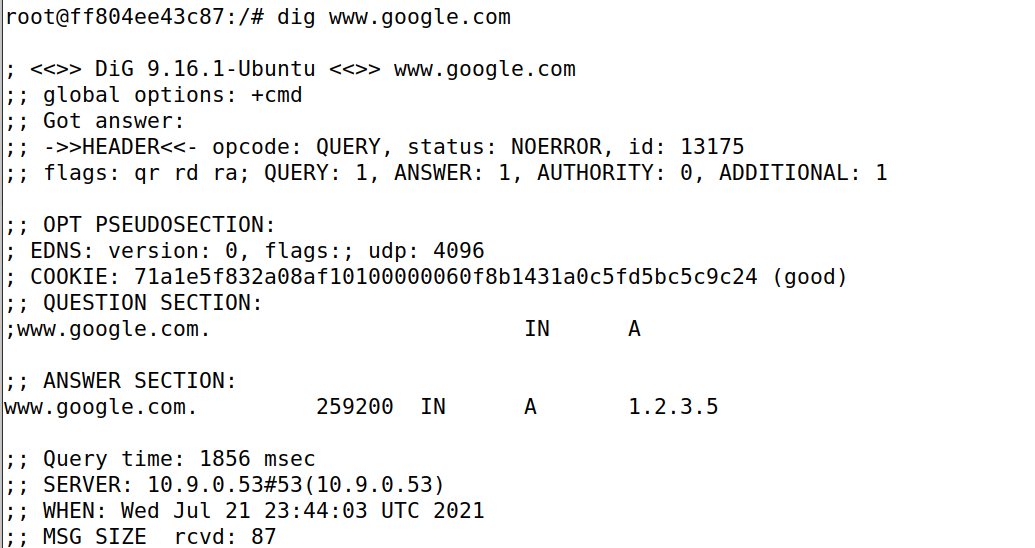
查看缓存，只有example.com的映射关系被写入到缓存中：

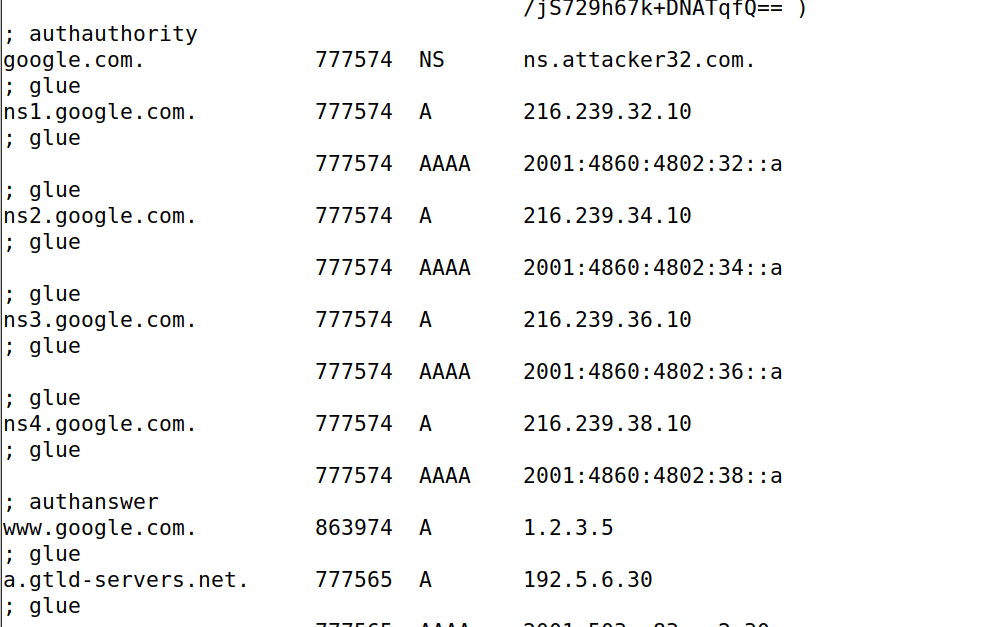




将DNS设置参数改成如下：ns=NSsec2/NSsec1，为了验证结果，我们对代码进行修改，NS选取[www.google.com](http://www.google.com)。结果如图，google.com到attacker32.com的映射被存储到缓存中



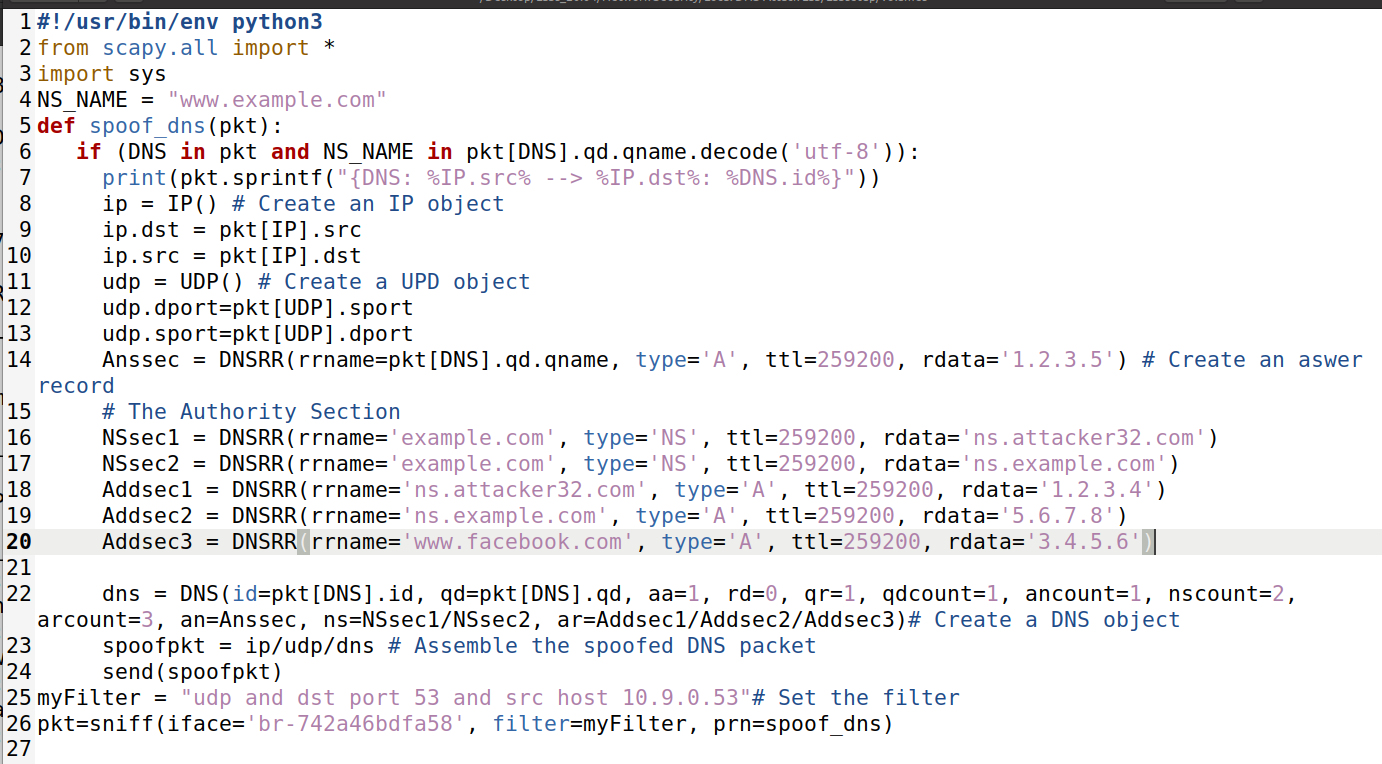




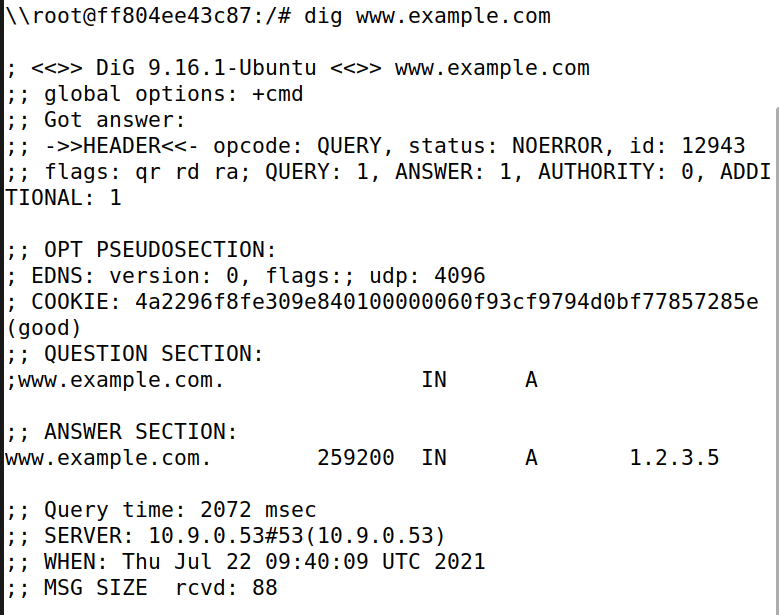
通过对比，发现前后并未发生变化，只有在响应报文中出现的记录才会被存储在缓存中。以此说明，缓存中只会存储NS选择的数据，其他域名不会出现在缓存中。

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

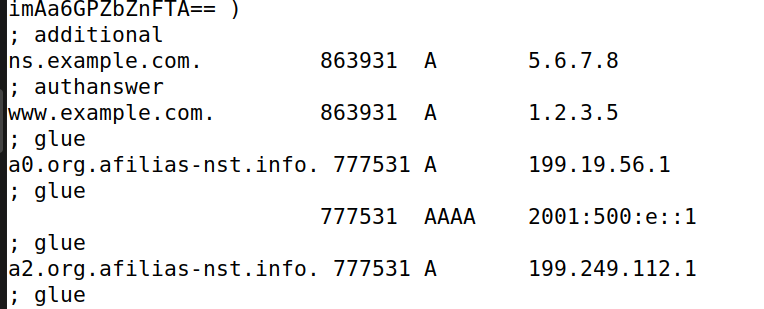
运行代码如下：



运行结果如下：



查看缓存，结果如下:



Additional section段只有ns.example.com被写入dns缓存中