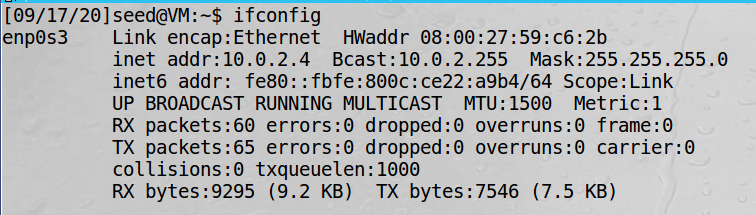
**网络安全综合课程设计实验５**

57117204 姚晓

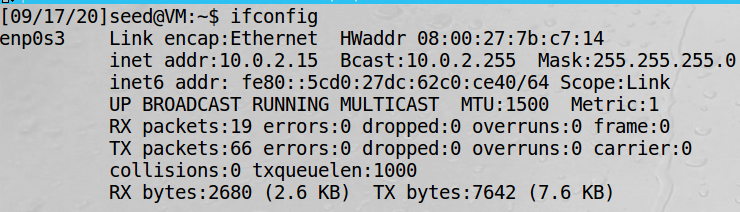
2020/9/12

**Lab Tasks: Setting Up a Local DNS Server**

本次实验用了三台虚拟机，令Ubuntu01（IP地址为10.0.2.4）为客户机：



令Ubuntu02（IP地址为10.0.2.15）为DNS服务器：

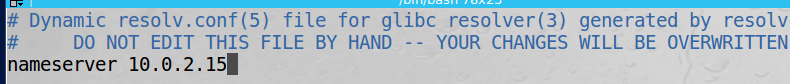


令Ubuntu03（IP地址为10.0.2.5）为攻击者：

**Task1: Configure the User Machine**

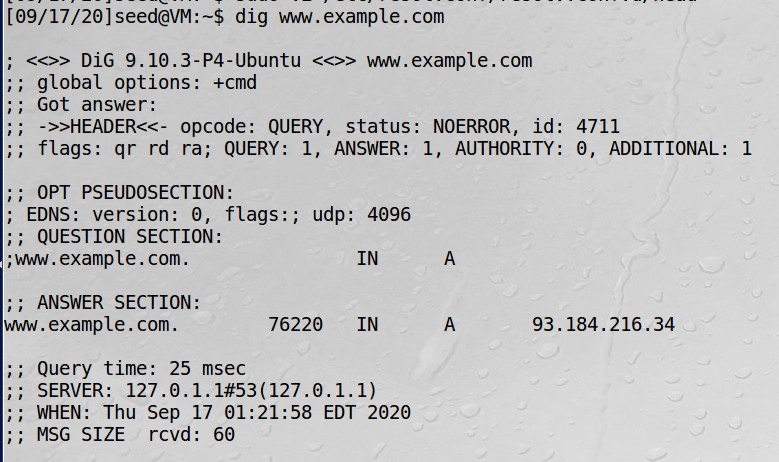
首先在/etc/resolvconf/resolv.conf.d/head文件里添加我们的DNS服务器的IP地址：



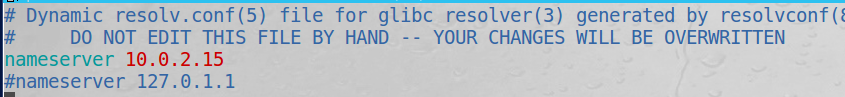




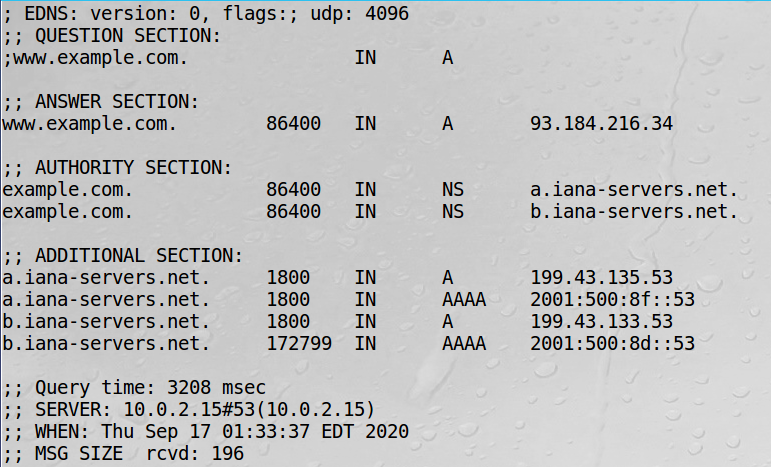
　　用dig命令查看当前的DNS服务器，发现不是我们设置的：



在/etc/resolv.conf文件中把ｎａｍｅｓｅｒｖｅｒ　１２７．０．０．１注释掉：

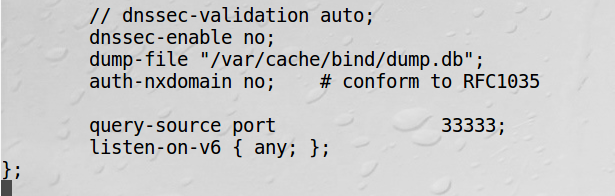


再用ｄｉｇ命令查看，发现本地DNS服务器变成了10.0.2.15，成功：

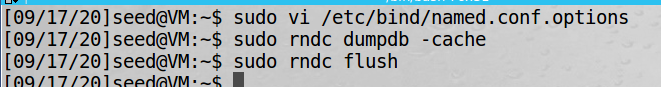


**Task2: Setup a Local DNS Server**

修改/etc/bind/named.conf.options文件，首先设置一个与DNS缓存有关的选项，并且关掉DNSSEC：



然后用下面两个命令，首先将cache里的内容转储到上面指定的文件中，然后清除高速缓存：

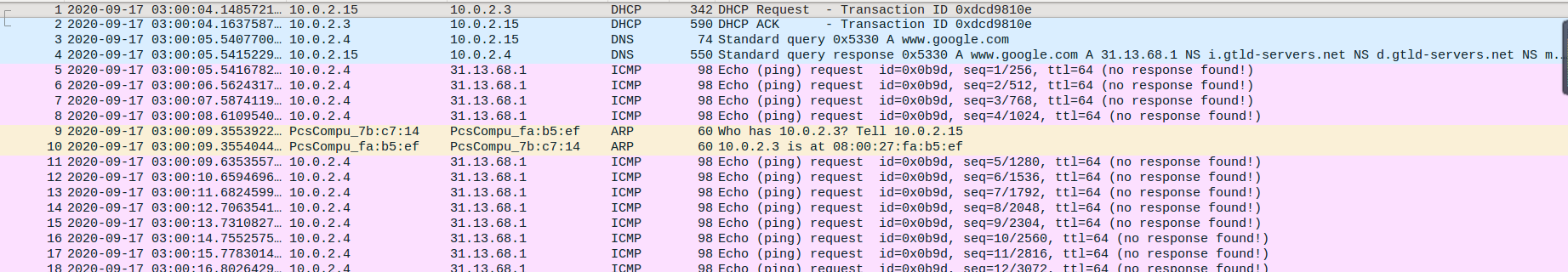


然后重启BIND9　DNS服务器：



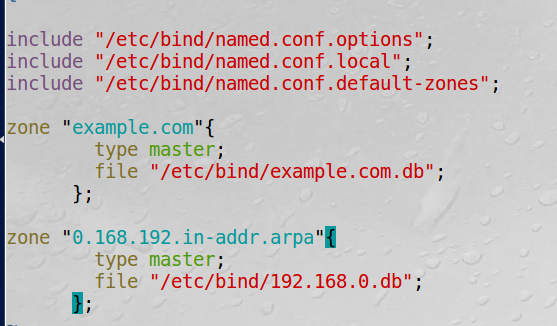
　　然后在客户机中ping　[www.google.com](http://www.google.com)，并用wireshark抓包，发现客户机先向１０.０.２.１５查询：



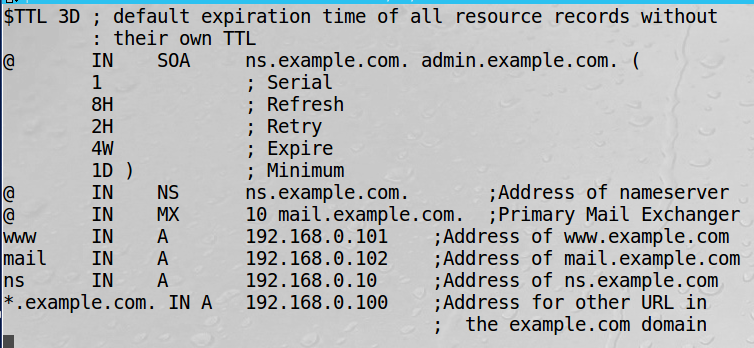


**Task３：Host　a　Zone　in　the　Local　DNS　Server：**

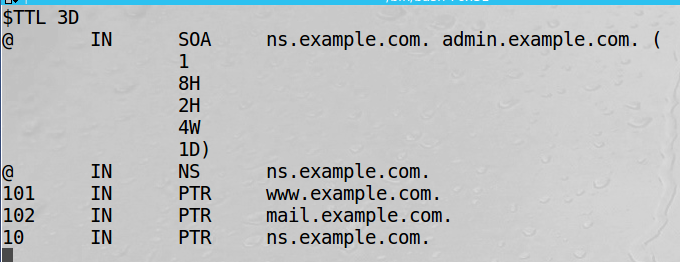
首先修改/etc/bind/named.conf文件，建两个zones：



然后设置正向查找区域文件。 上面区域定义中file关键字之后的文件名称称为区域文件，这是存储实际DNS解析的位置。 在/ etc / bind /目录中，创建以下example.com.db区域文件：



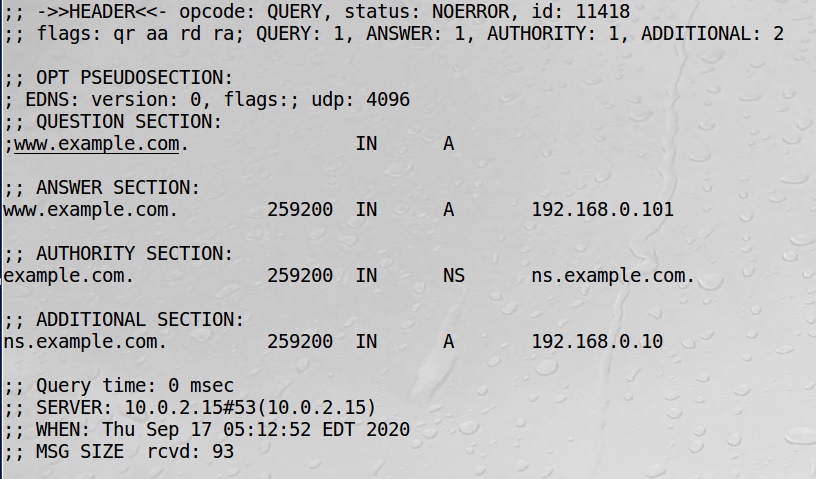
然后设置反向查找区域文件。 为了支持DNS反向查找，即从IP地址到主机名，我们还需要设置DNS反向查找文件。 在/ etc / bind /目录中，为example.net域创建以下反向DNS查找文件，称为192.168.0.db：



之后重启BIND：

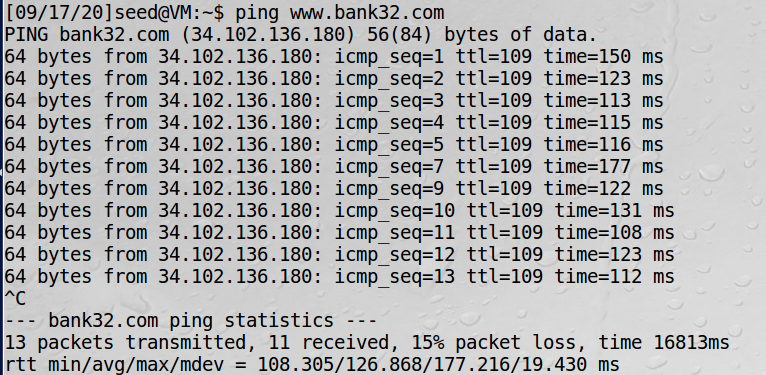


然后在客户机上用dig命令向本地DNS server问[www.example.com的IP](http://www.example.com的IP)地址，得到如下结果：

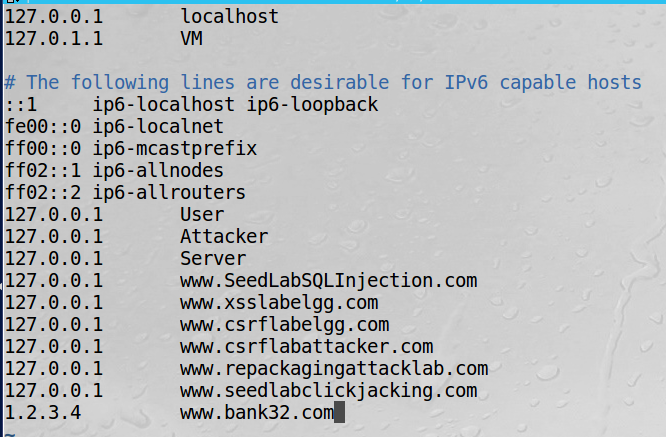


**Task4: Modifying　the　Host　File**

首先在客户机ping [www.bank32.com](http://www.bank32.com)：



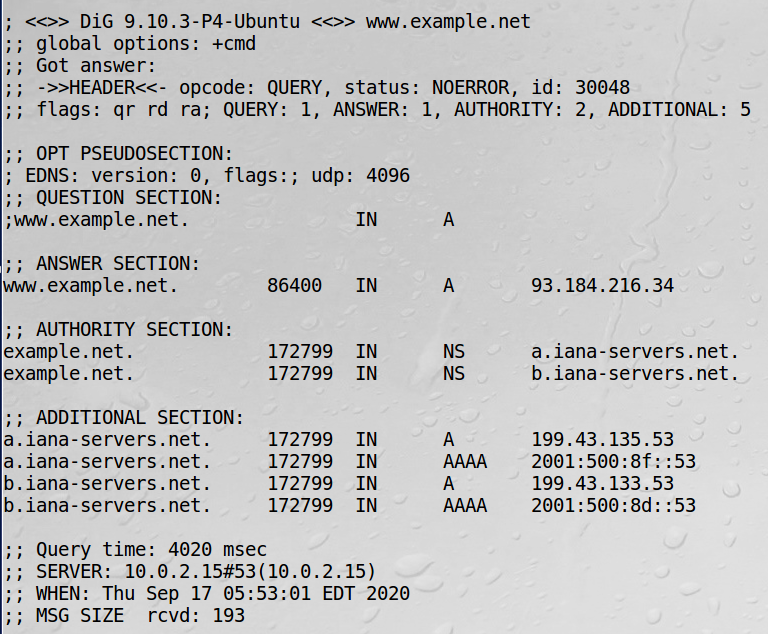
然后修改/etc/hosts文件，将[www.bank32.com重定向到1.2.3.4](http://www.bank32.com重定向到1.2.3.4)：



再次在客户机ping [www.bank32.com](http://www.bank32.com)，结果如下，不能ping通：

**Task5: Directly　Spoofing　Response　to　User**

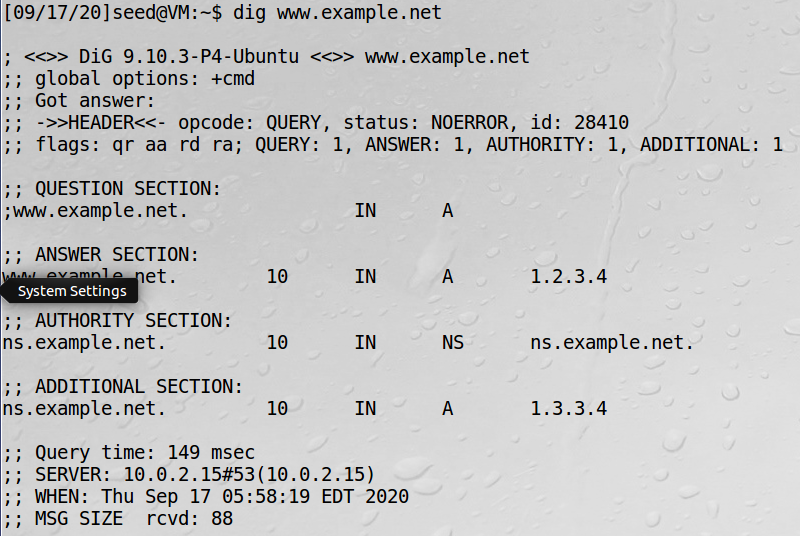
首先在客户机使用dig [www.example.net](http://www.example.net):

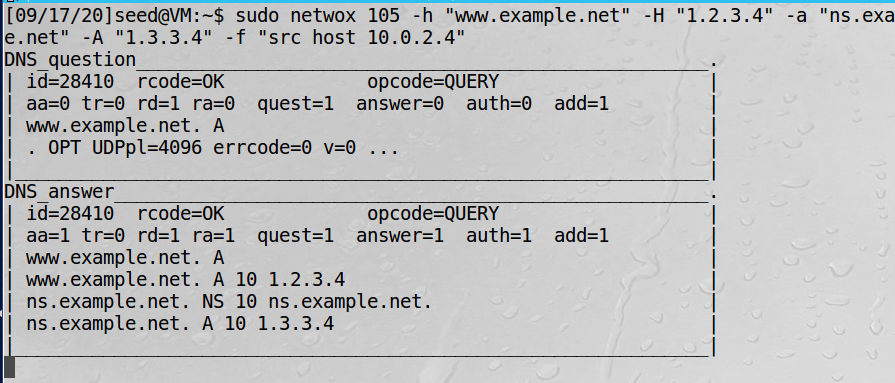


在攻击者的主机上用netwox嗅探请求报文，然后伪造一个回复报文：



在攻击的途中，在用户机dig [www.example.net](http://www.example.net)，得到如下结果，攻击成功，在回复消息里有伪造的欺骗信息，dig命令问到的DNS服务器的IP地址也是伪造的IP地址：





**Task6: DNS　Cache　Poisoning　Attack**

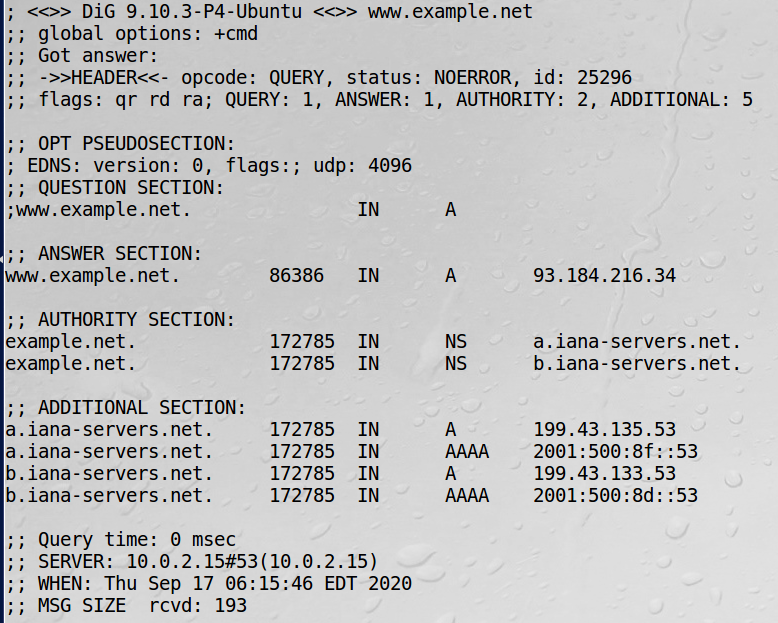
　　首先清除本地DNS服务器的缓存：

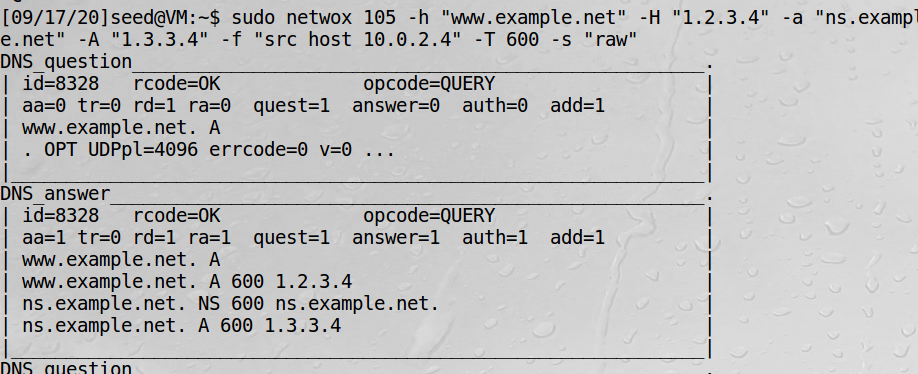


　　在攻击者的主机里用netwox伪造来自伪造来自其他DNS服务器的报文进行DNS cache poisoning攻击：



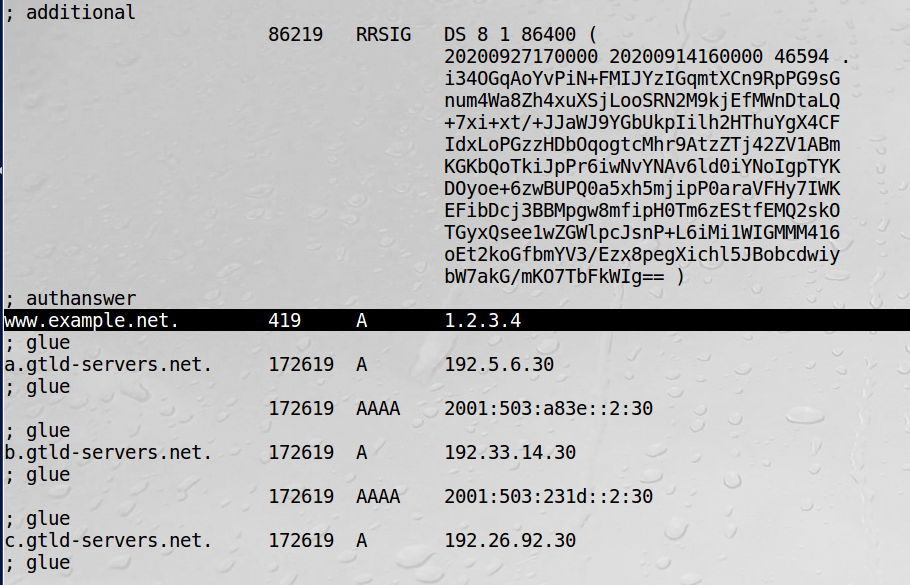
　　然后在用户机上dig [www.example.net](http://www.example.net)，得到入下结果:





　　在本地DNS服务器上查看缓存，找到结果：



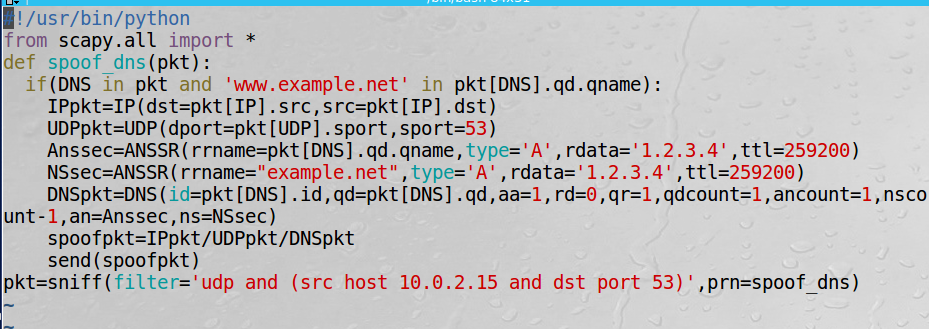


**Task7: DNS　Cache　Poisoning: Targeting　the　Authority　Section**

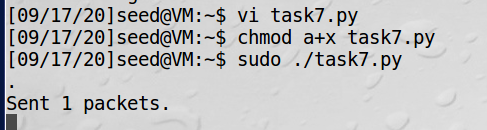
首先清除DNS服务器的缓存：



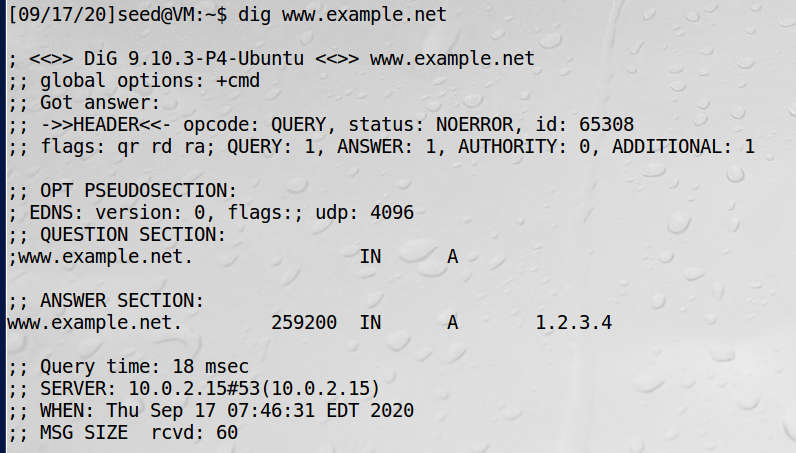
然后在攻击者的主机上构造攻击代码task7.py:



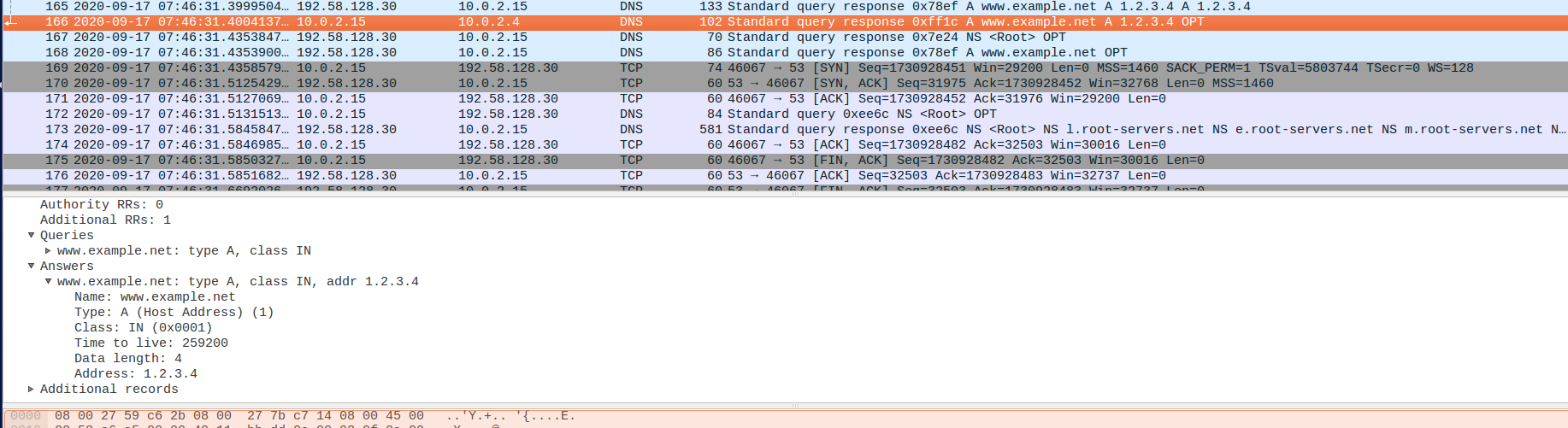
编译并运行代码：



在用户主机dig [www.example.net，返回的Answer为1.2.3.4](http://www.example.net，返回的Answer为1.2.3.4)：



Wireshark抓包结果如下：



然后在用户主机dig mail.example.net,并没有返回IP，但可以从wireshark抓包结果中看出攻击成功，ns.example.net也一样:

