**Lab-report5**

**Local DNS Attack Lab**

**Name:范心宇**

**Student number:57117129**

实验虚拟机：虚拟机A：10.0.2.5，客户机；

虚拟机B：10.0.2.6，服务器；

虚拟机M：10.0.2.4，攻击者。

**Lab Tasks (Part I): Setting Up a Local DNS Server**

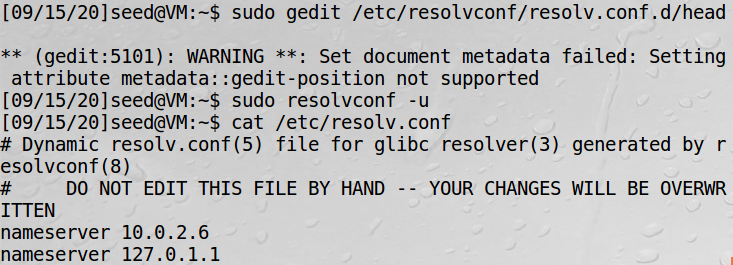
**Task 1: Configure the User Machine**

* **实验目的**

配置DNS客户主机。

* **实验过程**

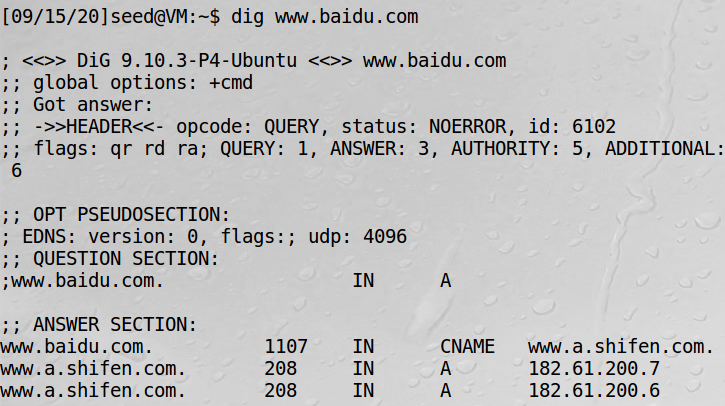
配置如下：

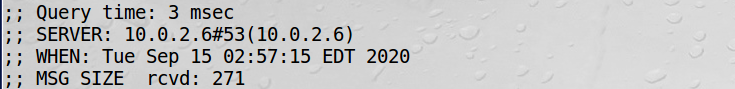




可以看到此时第一个nameserver域名服务器IP地址为10.0.2.6。

使用dig测试：





可以看到最后的SERVER字段，即DNS服务器，使用的是10.0.2.6，配置成功。

**Task 2: Set up a Local DNS Server**

* **实验目的**

配置本地DNS服务器。

* **实验过程**

1. 指定dump file。



1. 关闭DNSSEC。



1. 重启bind9服务。

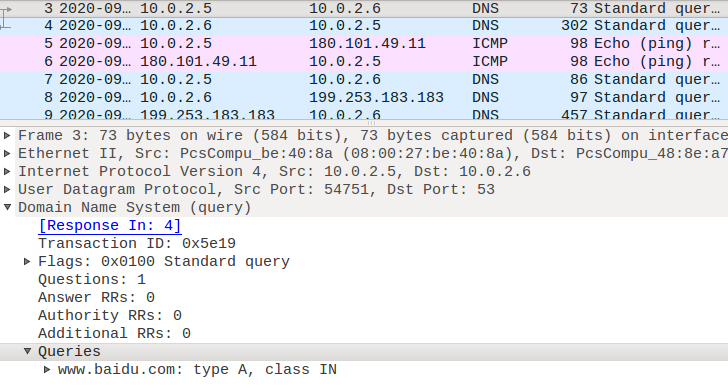


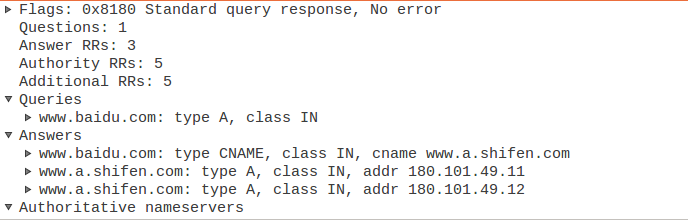
1. 使用DNS服务器。

在用户机A上使用ping命令：



观察wireshark：





可以看到虚拟机A向虚拟机B发起了DNS查询请求，得到www.baidu.com的IP地址后才发出PING报文。

**Task 3: Host a Zone in the Local DNS Server**

* **实验目的**

在本地DNS服务器上增加一个zone域。

* **实验过程**

1. 创建zone。

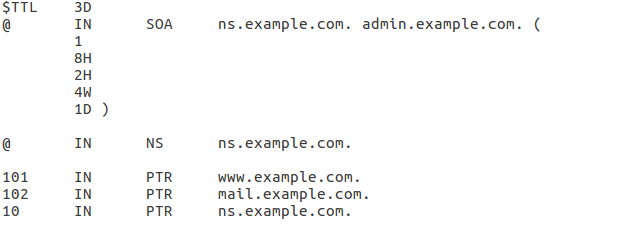


第一个“example.com”是正向域名解析，从DN到IP，第二个“0.168.192.in-addr.arpa”是逆向解析，从IP到DN。

1. 创建正向解析zone文件。

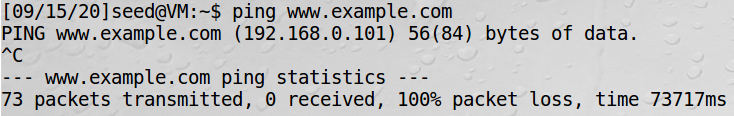


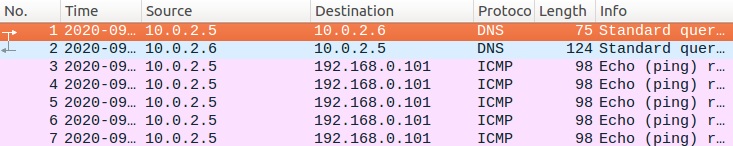
1. 创建逆向解析zone文件。



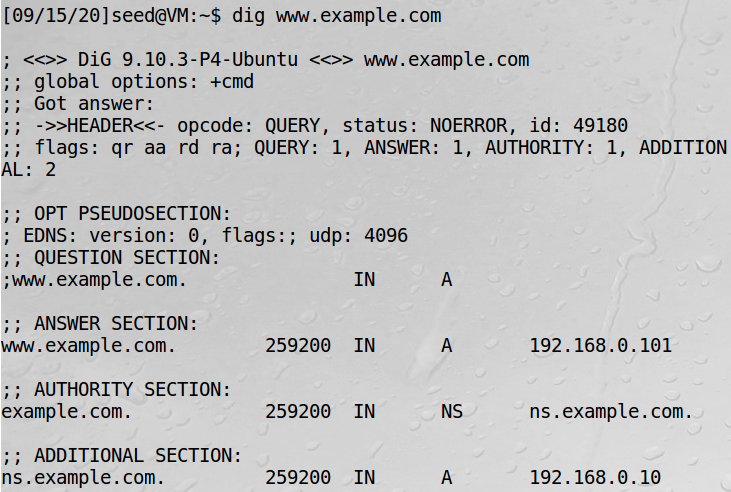
1. 重启DNS服务器程序并测试。

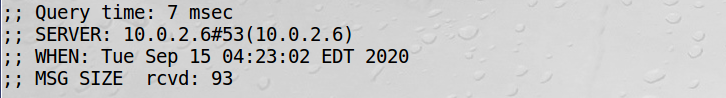






可以看到客户机向服务器发起了DNS请求，得到www.example.com的IP地址192.168.0.101，但由于是错误的IP地址，所以并不能ping通。





从使用dig得到的结果来看，客户机确实得到了服务器中存储的www.example.com的映射信息，获得IP地址192.168.0.101，服务器配置成功。

**Lab Tasks (Part II): Attacks on DNS**

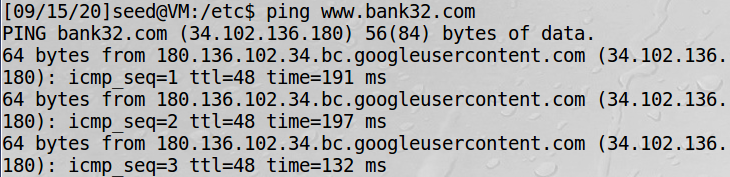
**Task 4: Modifying the Host File**

* **实验目的**

修改host文件，观察主机访问域名结果的变化。

* **实验过程**

1. 修改前ping [www.bank32.com。](http://www.bank32.com。)

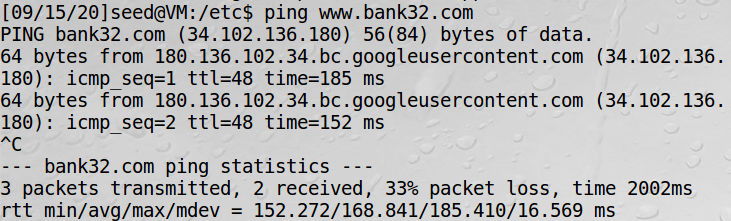


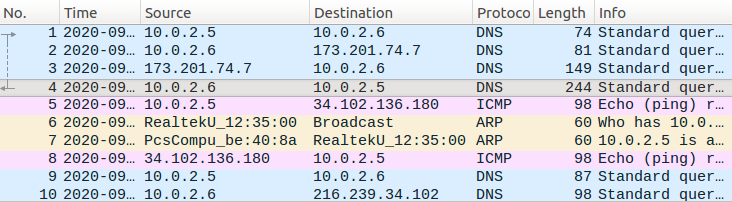
可以看到www.bank32.com的IP地址为34.102.136.180。

1. 修改后ping [www.bank32.com。](http://www.bank32.com。)

随意设置IP地址：

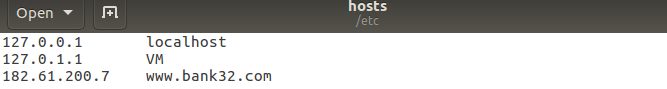




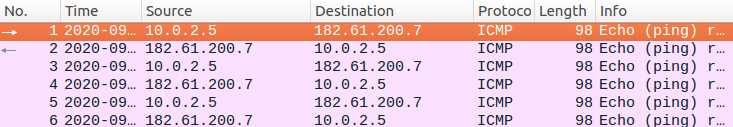


可以看到此时ping的仍然是www.bank32.com的正确IP地址，观察wireshark发现在发出ICMP报文前，首先发送了DNS请求报文，修正了IP地址。

使用百度的IP地址：







可以看到此时没有发送DNS请求报文，可以ping通，且ping的是百度的IP，也就是我们在hosts文件中设置的IP。

* **实验结论**

随意修改victim的/etc/hosts文件并不能达到重定向报文的目的，当设置的IP地址并不是存在的主机时，victim会发出DNS请求报文验证域名对应的IP，随即可以修正重定向项目。但如果设置的IP地址存在，刚好指向攻击者主机，那么victim就不会检查该解析结果是否正确而直接与attacker联系。

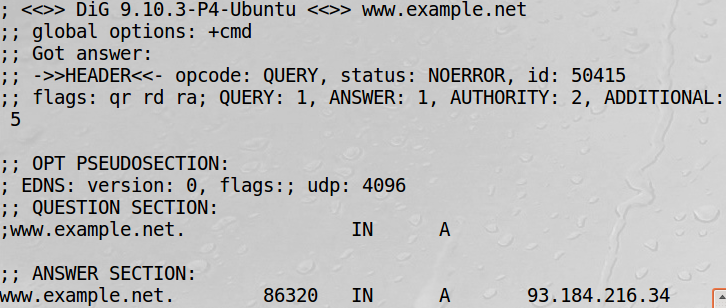
**Task 5: Directly Spoofing Response to User**

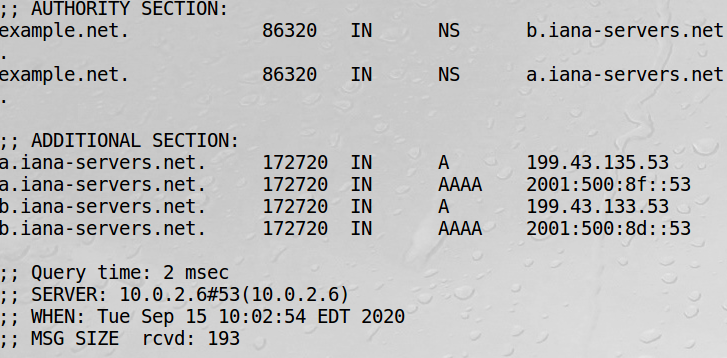
* **实验目的**

直接伪造DNS响应报文欺骗受害者。

* **实验过程**

1. 攻击前使用dig查看DNS信息。



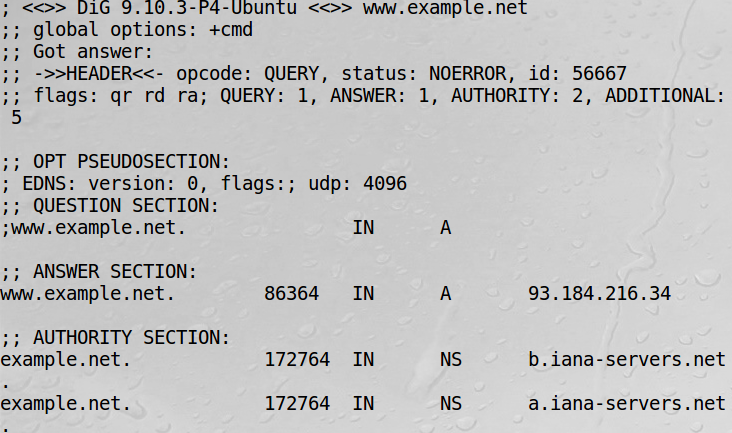


可以看到www.example.net的IP地址为93.184.216.34，信息来源于DNS服务器10.0.2.6。

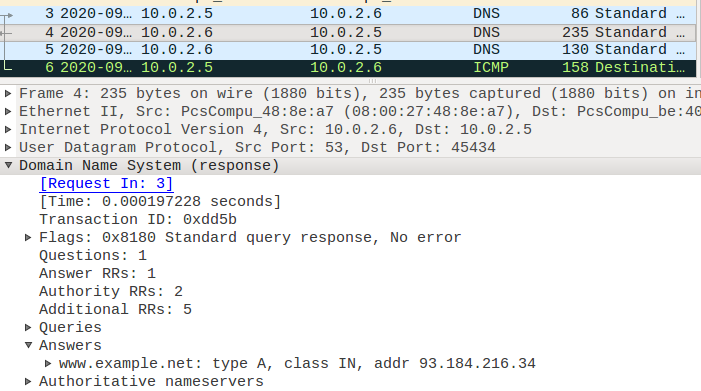
1. 攻击后使用dig查看DNS信息。



若DNS服务器曾查询过www.example.net的DNS信息：



可以看到客户端查询DNS信息后得到的仍然是正确IP，观察wireshark：

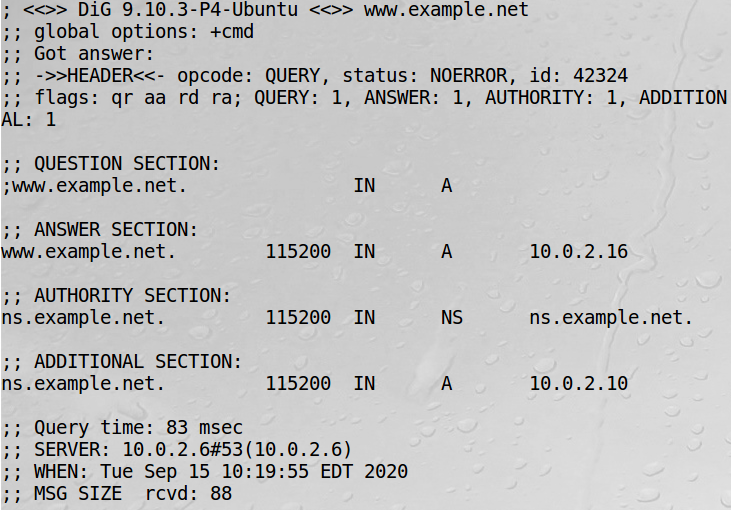


可以看到先到达的是真正DNS服务器给出的正确IP地址，攻击失败。

清理DNS服务器缓存：



再次发动攻击：



此时成功传递了错误的IP信息，攻击成功。

* **实验结论**

要想使直接针对客户端的伪造DNS报文攻击成功，必须要让伪造的报文在真正DNS服务器应答报文前到达客户端。在本实验中，如果DNS服务器事先已知了客户查询的域名对应的IP，netwox是来不及抢先一步传递信息的，只有当DNS缓存清空后，能够成功实施攻击。

**Task 6: DNS Cache Poisoning Attack**

* **实验目的**

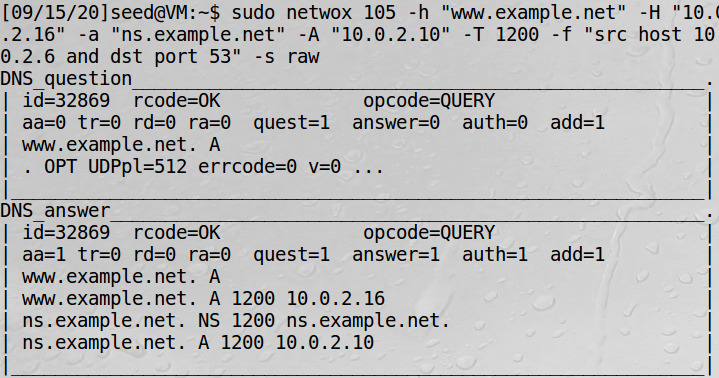
实施DNS缓存中毒攻击。

* **实验过程**

1. 清空DNS服务器缓存。



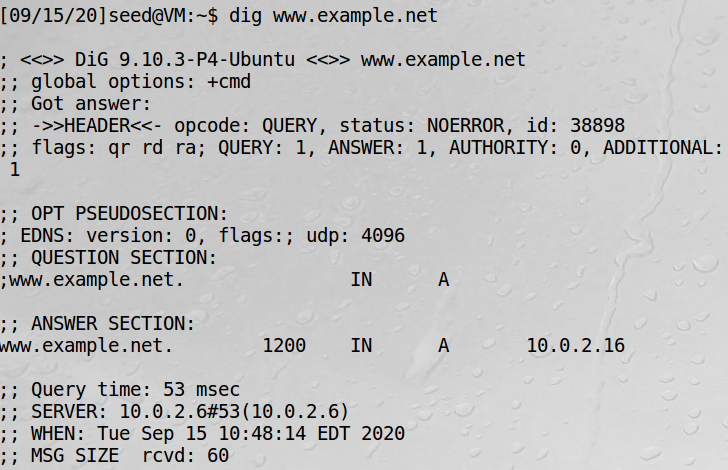
1. 实施攻击。



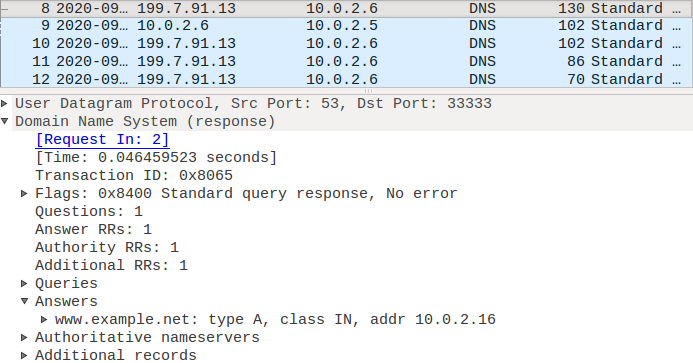
使用10.0.2.16作为伪造的IP地址，过滤器中源主机IP改为10.0.2.6。

1. 查看攻击结果。

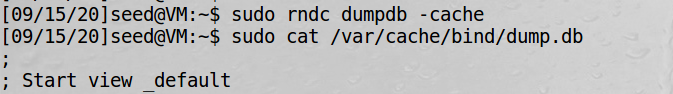
在客户端使用dig命令查询www.example.net的DNS信息：

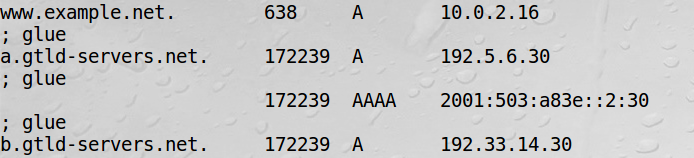


可以看到查询得到的结果是10.0.2.16，是我们伪造的IP地址。观察wireshark：



可以看到，伪造的DNS响应报文先到达本地DNS服务器，返回了我们伪造的IP地址10.0.2.16。再查看本地DNS服务器的cache内容：





可以看到出现的第一个IP地址就是www.example.net的IP地址，其值为10.0.2.16，是我们设置的攻击数值，攻击成功。

**Task 7: DNS Cache Poisoning: Targeting the Authority Section**

* **实验目的**

对整个zone进行DNS缓存中毒攻击，即将整个zone的权限服务器导向我们的服务器。

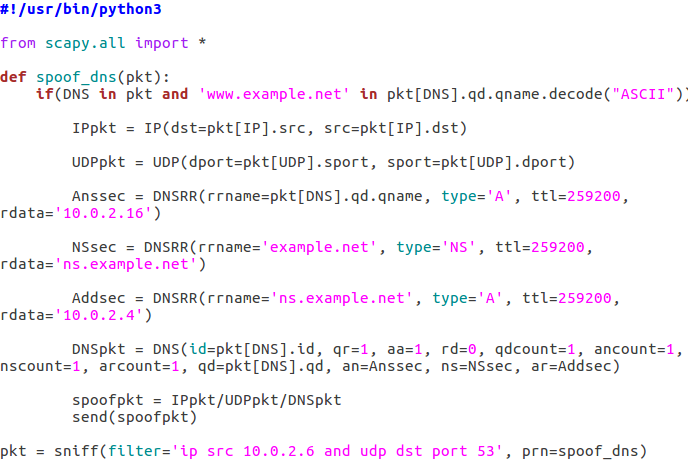
* **实验过程**

1. 清空本地DNS服务器缓存。

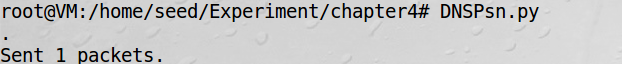


1. 编写攻击代码。

伪造DNS响应报文，使其zone的权限服务器导向我们的服务器：

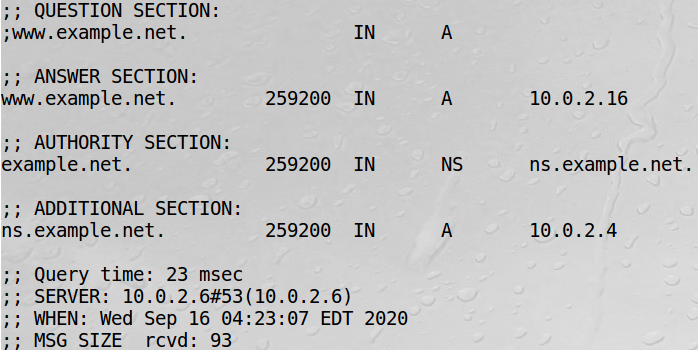


1. 执行攻击。



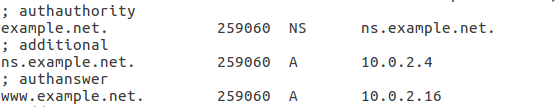
1. 观察攻击结果。

在客户端使用dig命令查询www.example.net对应IP：



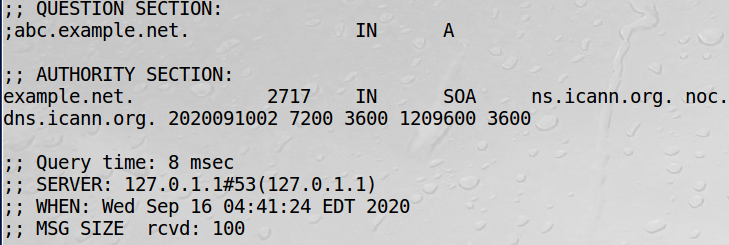
可以看到查询结果来自本地服务器10.0.2.6，查询结果为10.0.2.16，提供的example.net区域的权限服务器域名为ns.example.net，对应IP地址为10.0.2.4，是我们的服务器地址。

再查看本地DNS服务器Cache缓存情况：

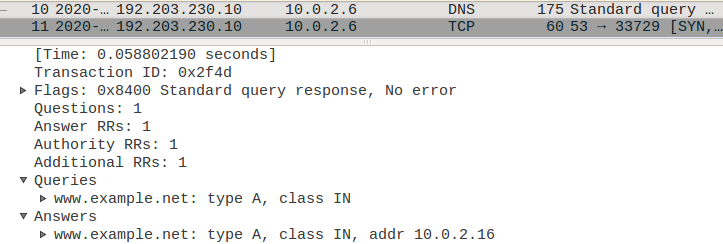


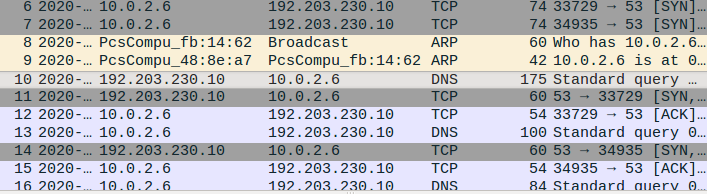
可以看到其记录example.net区域对应权限服务为ns.example.net，IP地址为10.0.2.4。

再使用dig命令查询其他example.net上的域名：



可以得到正确结果，观察wireshark：





我们伪造的报文确实传递给了本地DNS服务器，但同时本地DNS服务器与其发送请求的外部DNS服务器使用TCP进行了通话，



可以看到本地DNS服务器并未向我们的服务器发出DNS请求，而是选择了上面TCP对话中提供的根域名服务器。