**Lab5 Report**

**Local DNS Attack Lab**

57117107 赵笛言

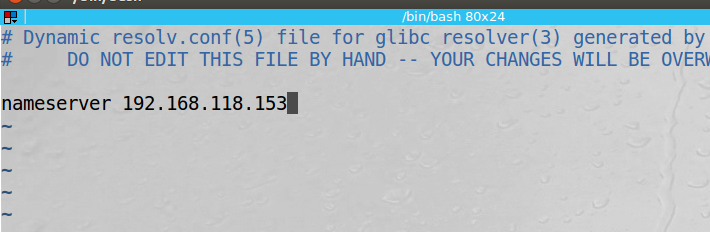
**LabTasks：**

Set：设置三个主机，分别作为用户机，DNS服务器，攻击者。IP地址分别为：192.168.118.151；192.168.118.153；192.168.118.150.

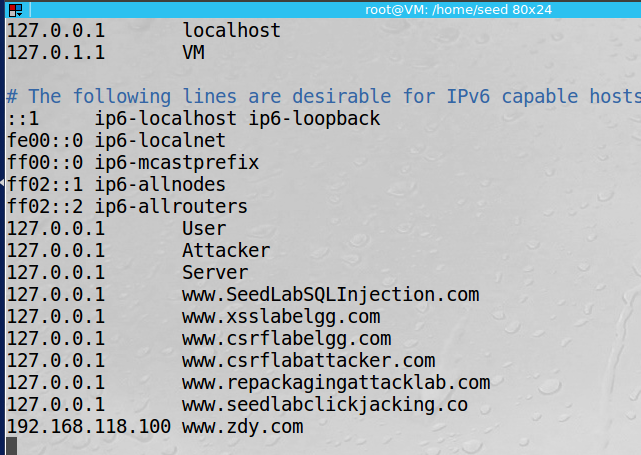
Task1: Configure the User Machine

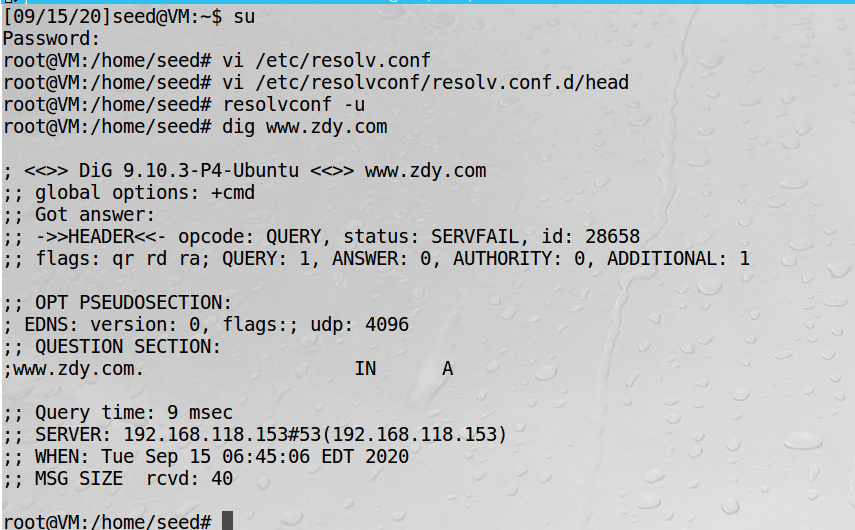
设置用户机（.151）：

在/etc/resolvconf/resolv.conf.d/head中加入nameserver信息



通过dig一个在DNS服务器（.153）里添加的网址，可以测试出添加成功。

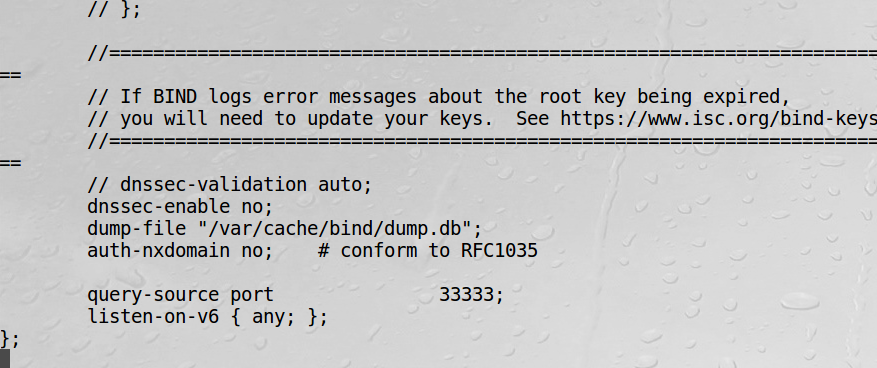




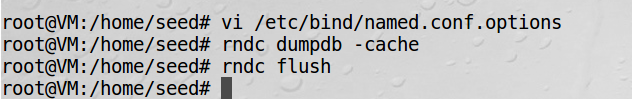
Task2: Setup a Local DNS Server

配置BIND9：

添加dump-file条目；关闭DNSSEC；使用固定源端口号；



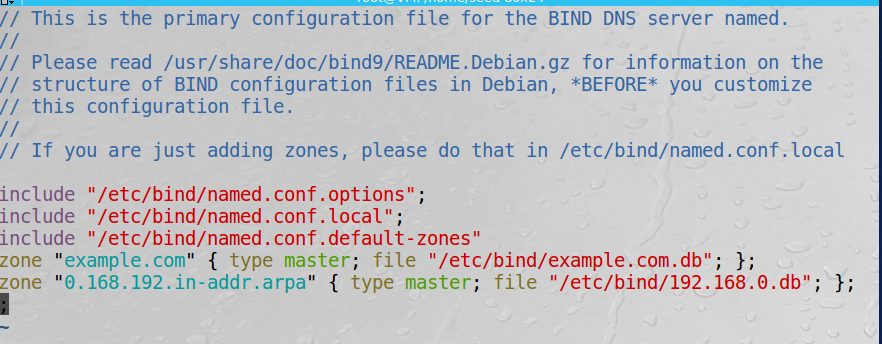
通过如下命令设置缓存储存文件、清空缓存。并重启DNS服务器。



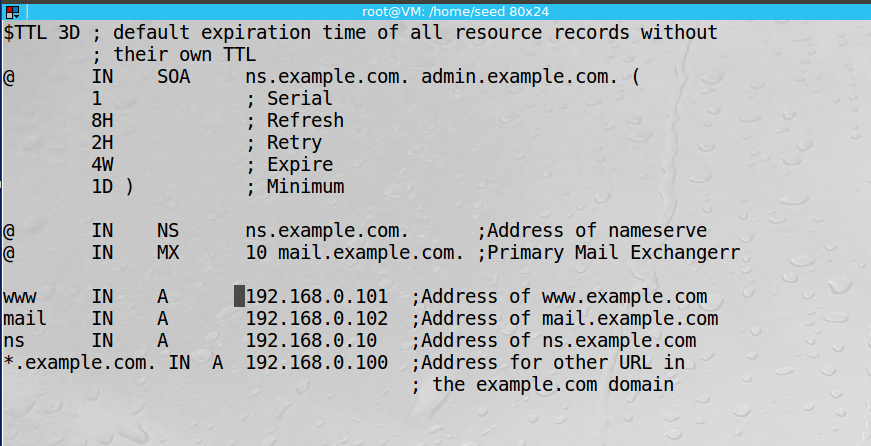
Task3: Host a Zone in the Local DNS Server

在本地DNS服务器中配置区域。

在/etc/bind/named.conf中新建区域:



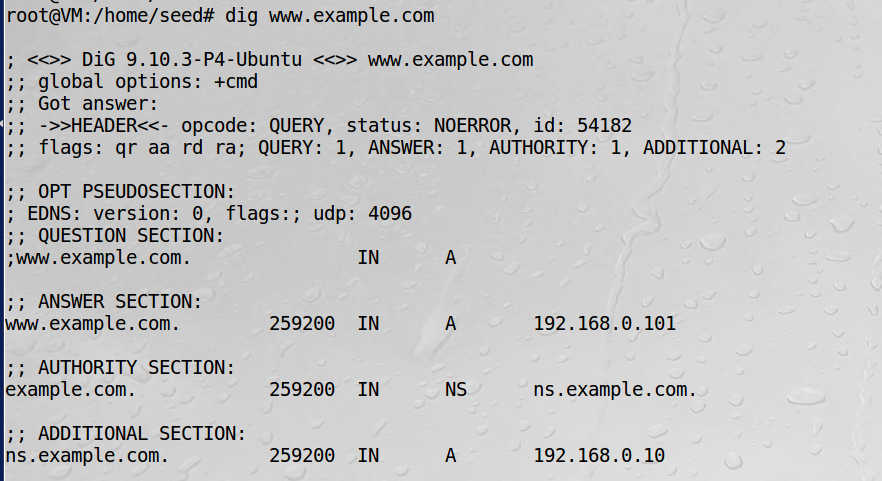
在/etc/bind/example.com.db配置正向查找区域文件:



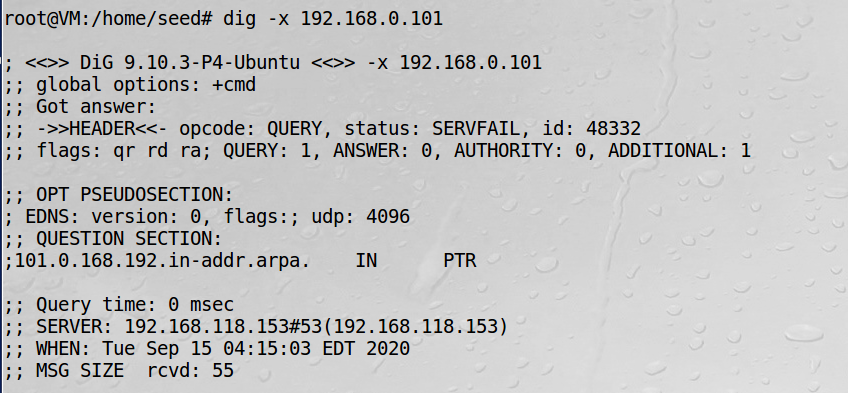
在/etc/bind/192.168.0.db配置反向查找区域文件:



重启服务器并测试：



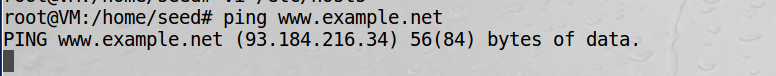
DNS反向解析：



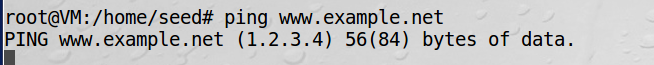
Task4: Modifying the Host File

直接修改用户机的/etc/hosts文件，可以将example.net重定向到恶意站点（此处为1.2.3.4）

攻击前：



攻击后：

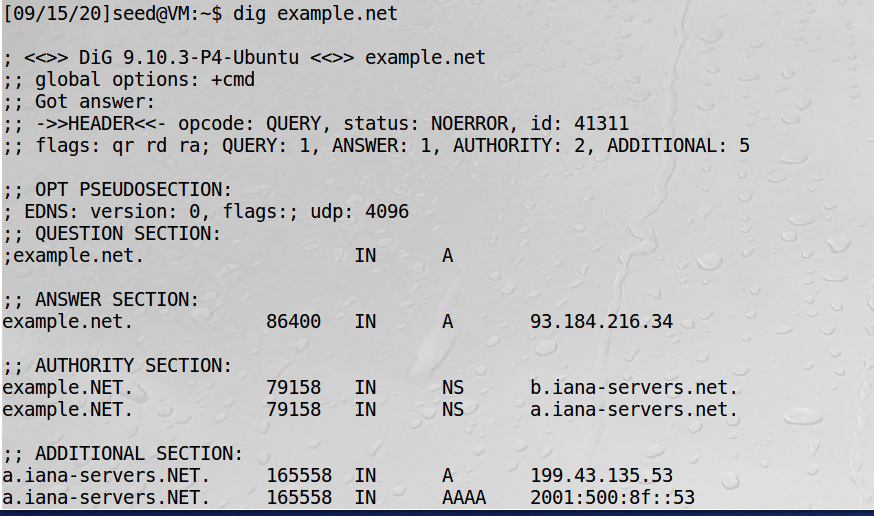


Task5: Directly Spoofing Response to User

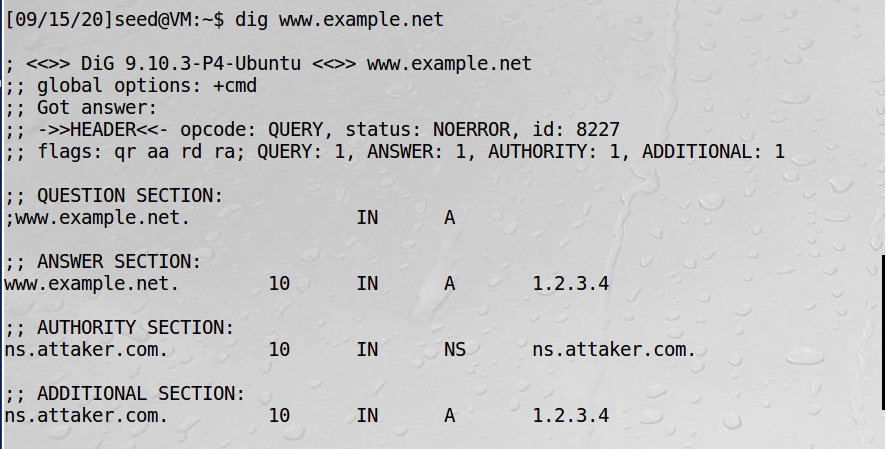
在攻击者（.150）上通过netwox工具105直接伪造对用户机（.151）网页访问的伪造应答：

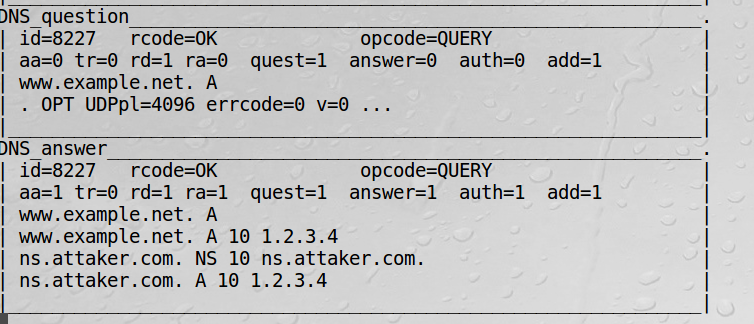


攻击前尝试dig example.net：



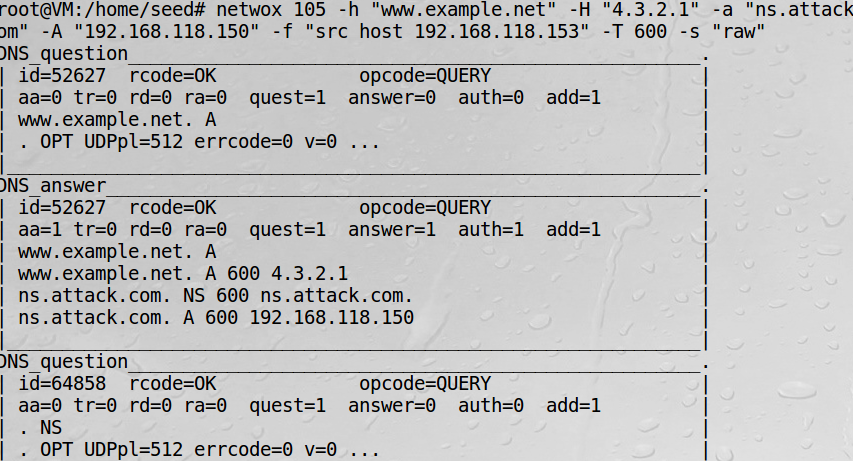
攻击后再次dig，可说明攻击成功：



而攻击者界面为：  


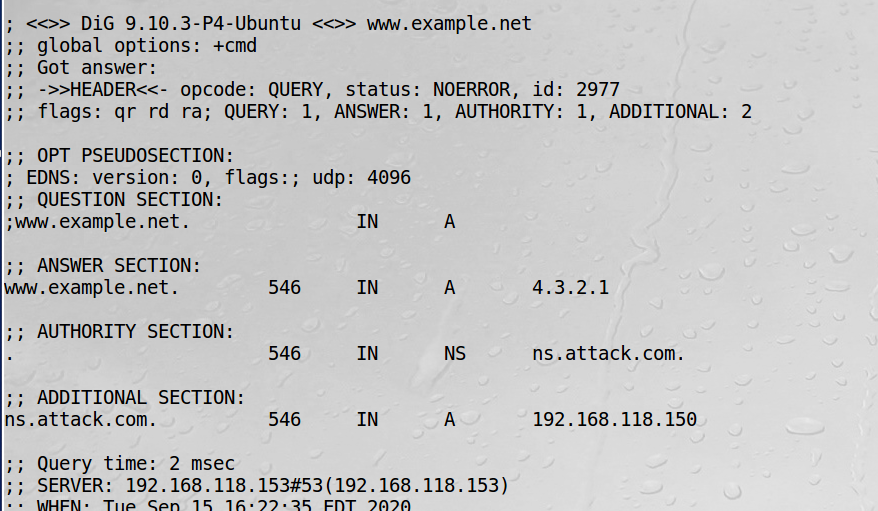
Task6: DNS Cache Poisoning Attack

此处不在对用户机进行短暂低效的攻击，而选择直接对DNS服务器进行可持续一段时间的攻击，依然使用netwox工具：

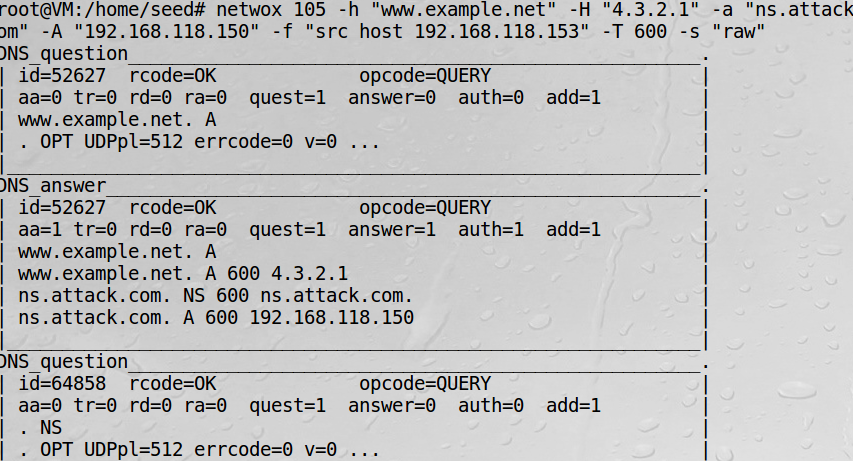


需要注意spoof设置为raw，以及攻击前对服务器缓存进行刷新处理避免上一次攻击的影响。（我是重启后才成功使缓存信息消除）

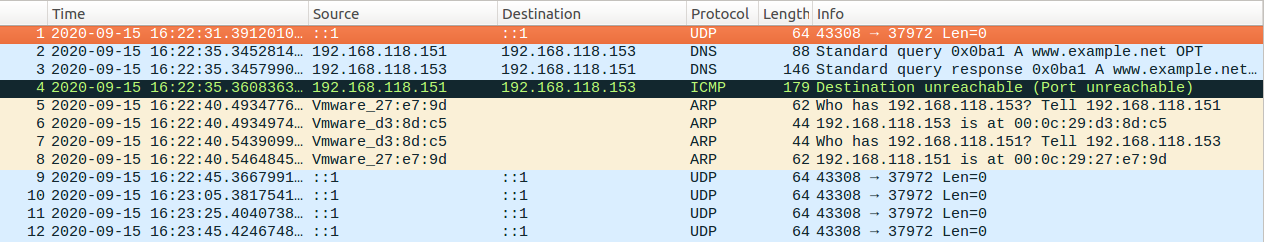
此时用户dig example.net，表示攻击成功：



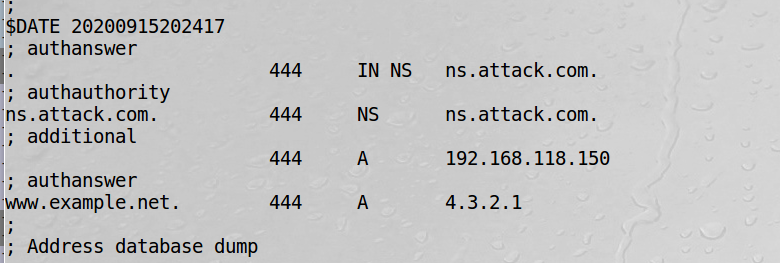
而攻击者界面：



此时wireshark的信息捕捉如下图：



再去检查转储的缓存，发现已被污染：

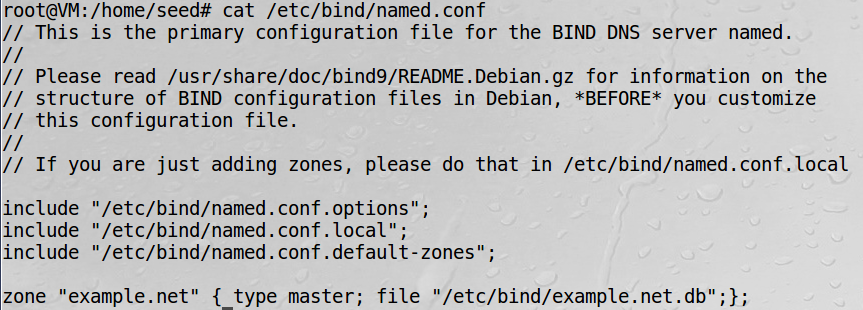


Task7: DNS Cache Poisoning: Targeting the Authority Section

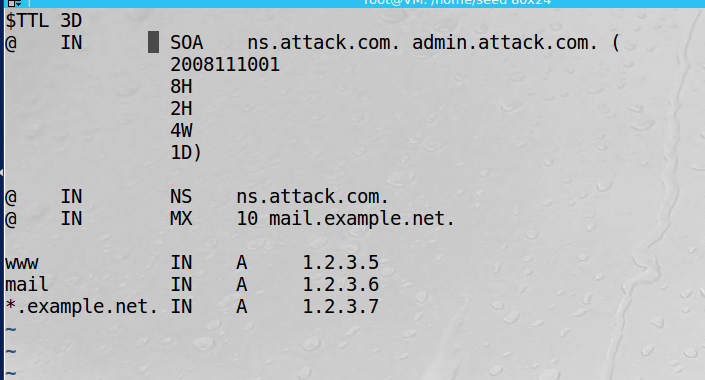
上述攻击方法成本过高，因此可以尝试通过以下攻击方法完全劫持example.net:

首先依然通过netwox命令伪造回复报文，与上述实验没有区别，需要保证ns.attack.com的IP地址是攻击者的IP地址，即192.168.118.150。

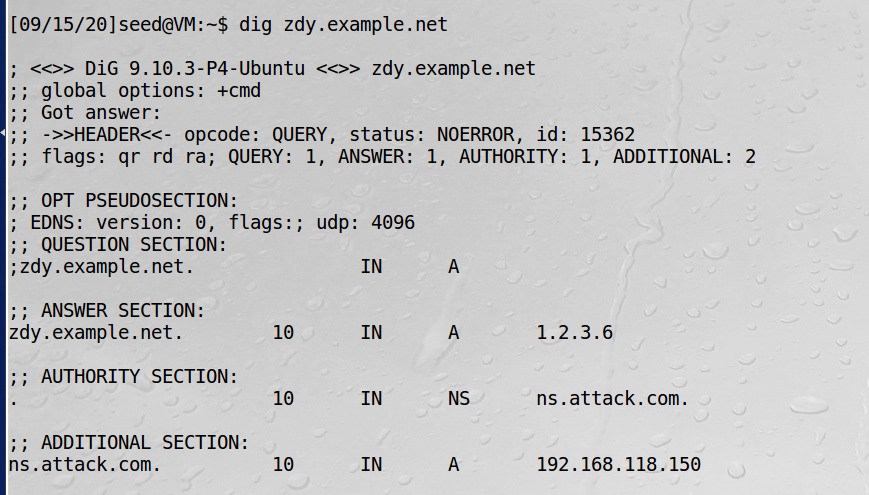
然后修改攻击者的/etc/bind/named.conf文件：



再添加区域文件/etc/bind/example.net.db:



在重启BIND服务器后，在用户机上对example.net域内任一主机访问，都会如下显示：



证实攻击成功。

总结：

实验总的来说不算很复杂，结合书上的原理来看较为清晰，但比起书上通过写py脚本来实现攻击，netwox攻击明显更简洁。

在实验进行中有多处需要DNS服务器重启和缓存清空的步骤，如果忽略掉就会导致攻击无法实现想要的结果。