DNS Relay实验报告

PB19111713钟颖康

基本框架使用助教提供的"DNS_Relay Python代码实例"

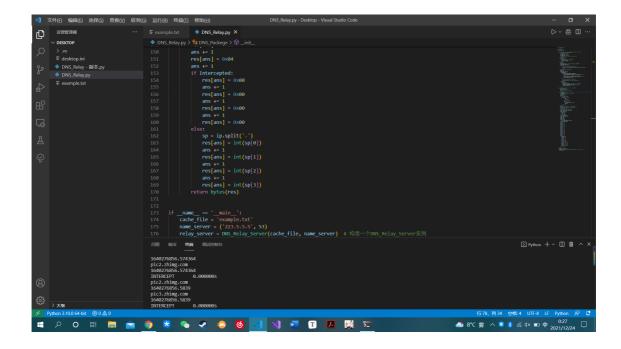
1.代码整体结构:

```
1 class DNS_Relay_Server: # 一个relay server实例,通过缓存文件和外部地址来初始化
2
      def __init__(self, cache_file, name_server): # 初始化
3
4
     def load_file(self, ):
                                                 #读取配置文件
5
                                                 #创建UDP连接,给每个
6
      def run(self):
   DNS报文创建一个非阻塞线程
7
       def handle(self, server_socket, data, addr): #处理DNS报文
8
9
10
11 class DNS_Packege: # 一个DNS Frame实例,用于解析和生成DNS帧
12
       def __init__(self, data):
                                                #从data中解析信息
13
14
       def generate_response(self, ip, Intercepted): #构造响应报文
15
```

2. 查询与响应的实现过程

- 。 完善了 DNS_Relay_Server 类中的 handle() 函数。处理 DNS报文的具体逻辑为:如果是查询报文,且如果网址在配置文件中有且是A类查询,则构建响应报文,其余网址或其余类型的查询则直接将报文转发到正常的 DNS Server;如果是响应报文且有相应的转发记录,则发送回之前保存的对应表项的源地址。
- o 其次,在 DNS_Packege 类中,结合实验文档"RFC1035"完善了___init___:按照固定模式从data中解析出需要的信息;generate_response 函数:构造响应报文的格式参见"RFC1035"。

3.程序运行结果



在上图的终端中可以看到知乎上的图片 (pic2.zhimg.com等) 被拦截 (INTERCEPT) , 具体效果如下:

